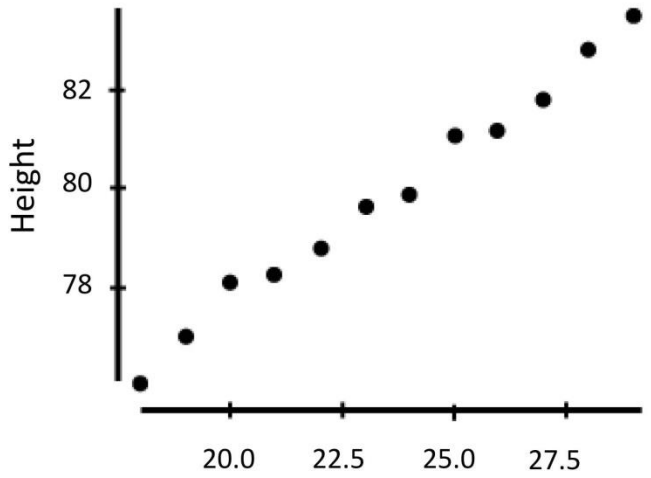
Correlation and Regression

- Correlation analysis is used for investigating the relationship between two quantitative variables
- Goals of correlation analysis :
  - Analyze if two measurement variables have a relation. This means change in one influences change in the other measure
  - Quantify the strength of the relationship between the variables

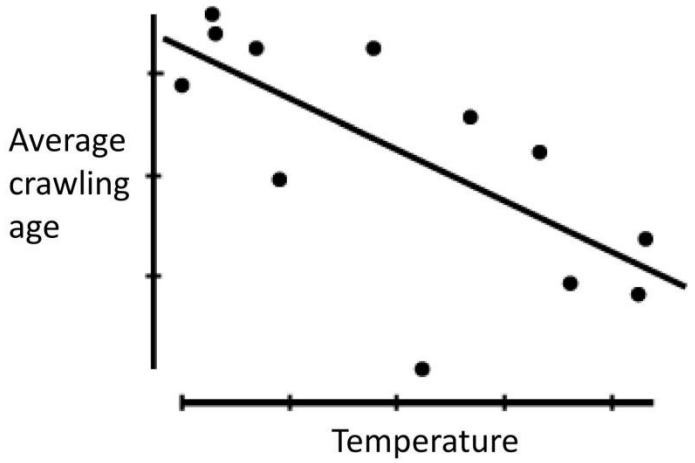
Correlation Examples

A positive correlation between height of a child and age: As the child grows his or her height increases almost linearly.

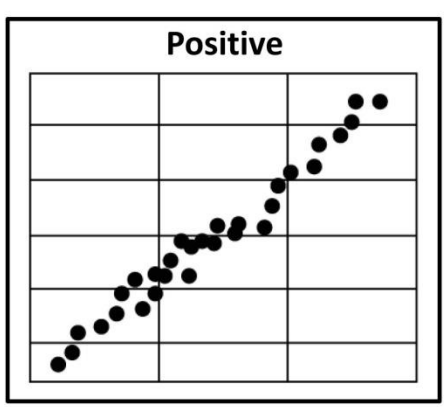


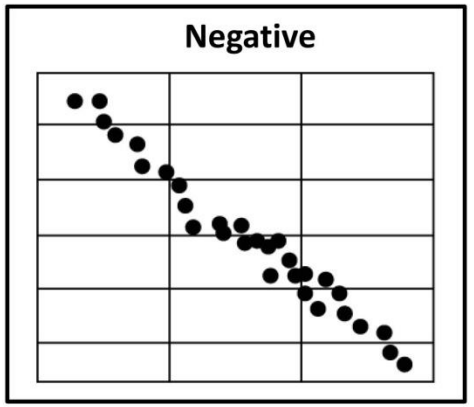
Correlation Examples (Cont'd)

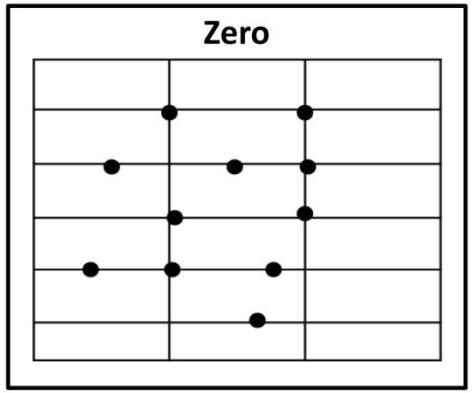
A negative correlation between temperature and time babies take to crawl: Babies take longer to learn to crawl in cold months (when they are bundled in clothes that restrict their movement), than in warmer months.



Correlation Coefficient (Cont'd)







Correlation and Regression Examples

Analysis of Student Grades in Mathematics and English

- Use Correlation to determine if the students who are good at Mathematics tend to be equally good at English
- Use Regression to determine whether the marks in English can be predicted for given marks in Mathematics

Correlation: Example

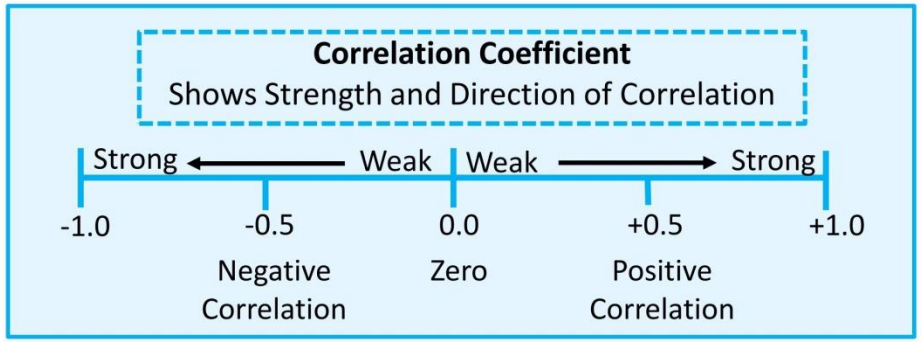
Correlation does not imply Causation.

- Myth: India team loses a match if Sachin Tendulkar hits a century
- Correlation: Sachin hits a century and India wins a match
- Does it imply causation? Does Sachin hitting a century causes India to lose the match?

Correlation Coefficient (Cont'd)

The Correlation Coefficient ranges from -1 to 1.

- +1 indicates perfect collinearity, which means, if one value increases, the other also increases in the same proportion
- -1 indicates perfect negative collinearity, which means, if one value decreases, the other increases in the same proportion
- Zero indicates no relationship between the variables



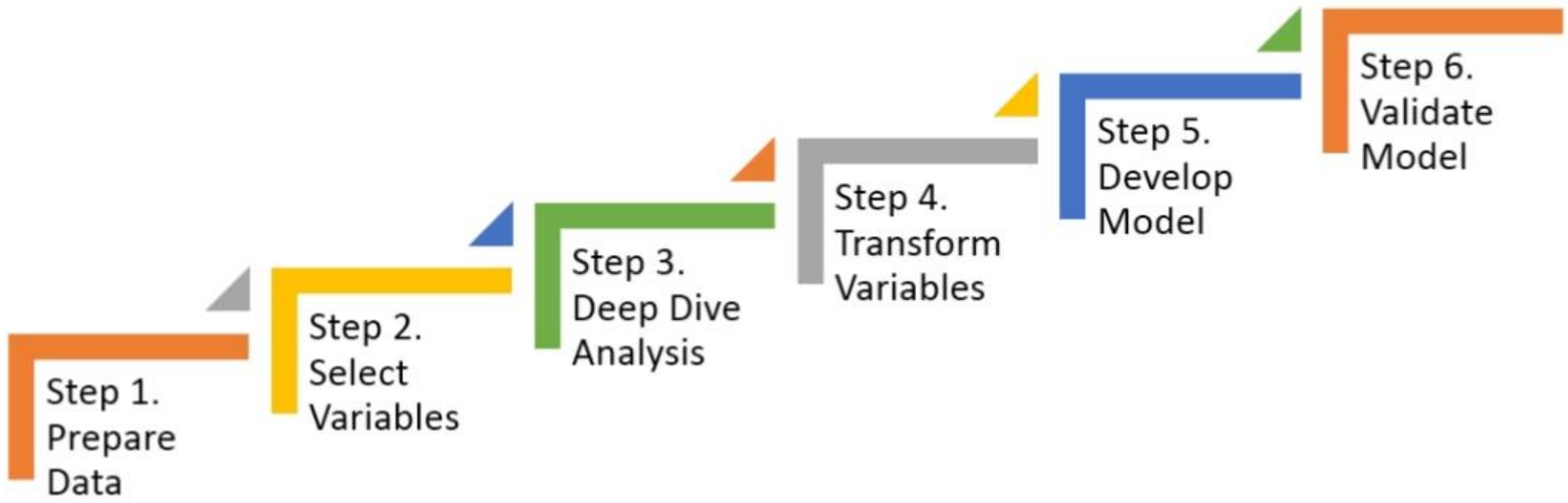
Regressions

In statistics, regression analysis is a statistical process for estimating the relationships among variables. ...

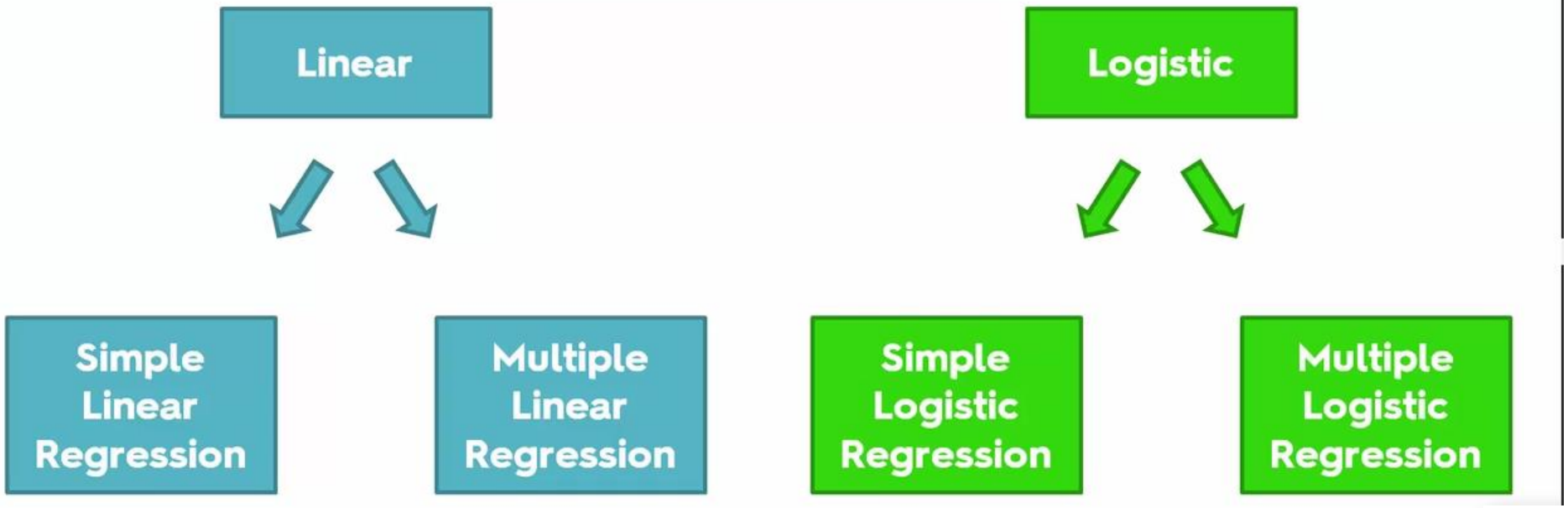
The focus is on the relationship between a dependent variable and one or more independent variables.

-Wikipedia

modeling process

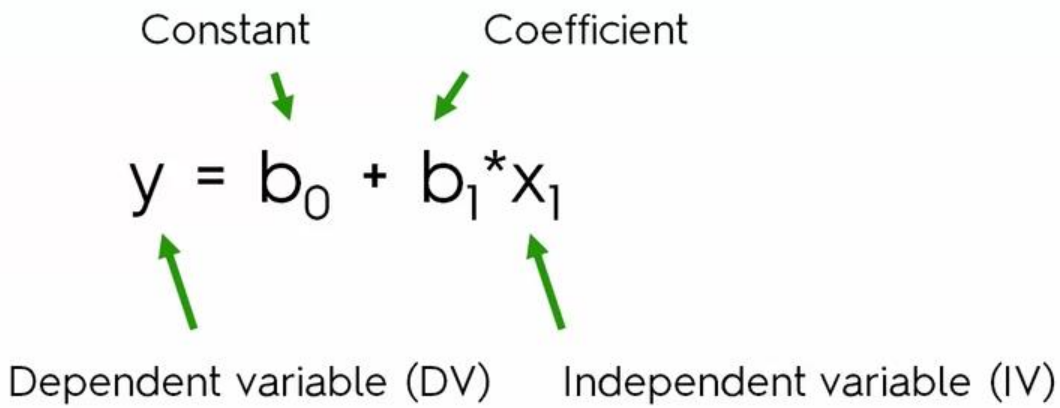


Regressions



Regressions

Simple Linear Regression



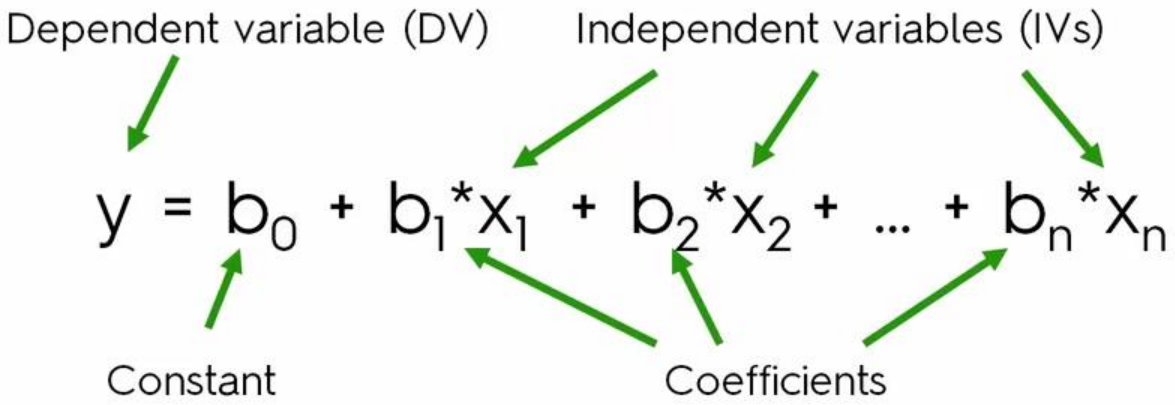
Multiple Linear Regression

Regressions

Simple Linear Regression

$$y = b_0 + b_1 * x_1$$

Multiple Linear Regression

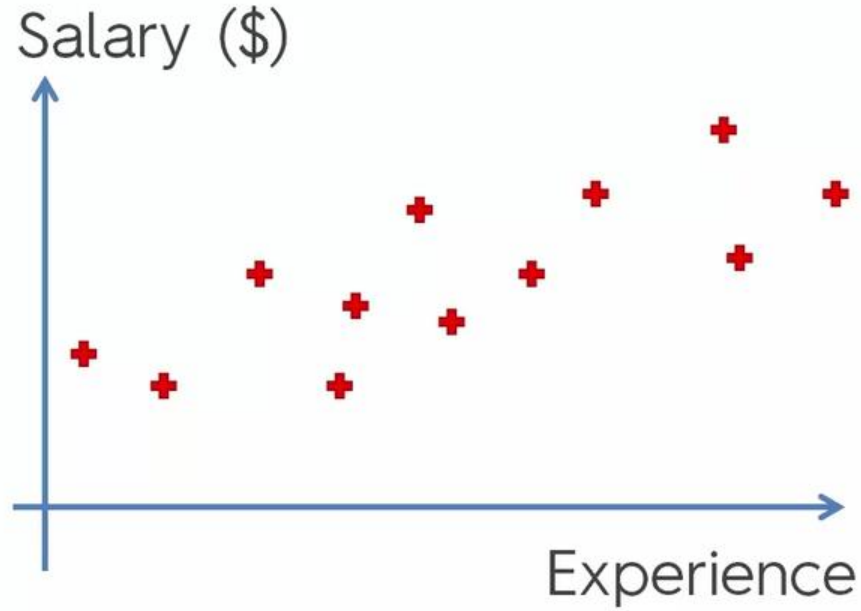


Regressions

Simple Linear Regression:

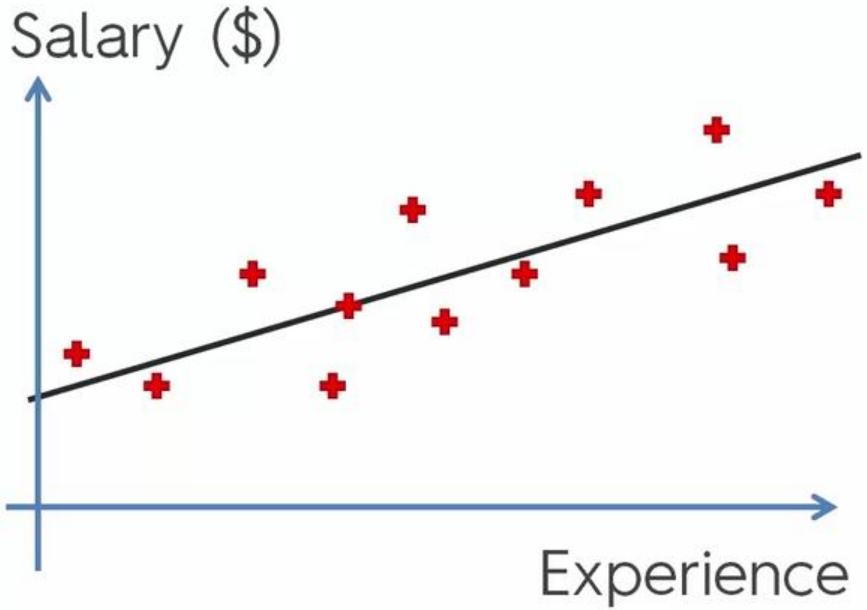
Regressions

Simple Linear Regression:



Regressions

Simple Linear Regression:

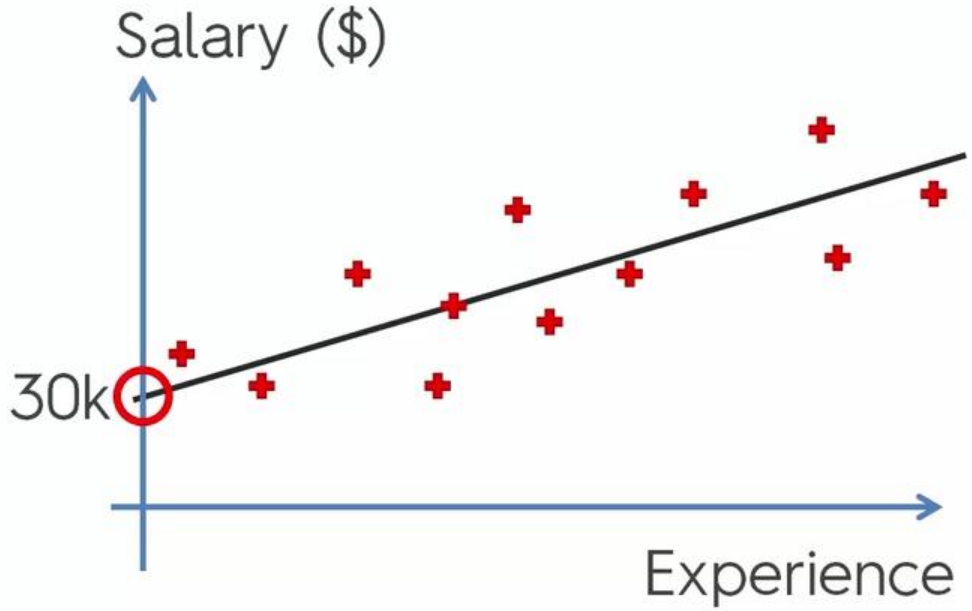


$y = b_0 + b_1 * x$

Salary = b_0 + b_1 *Experience

Regressions

Simple Linear Regression:

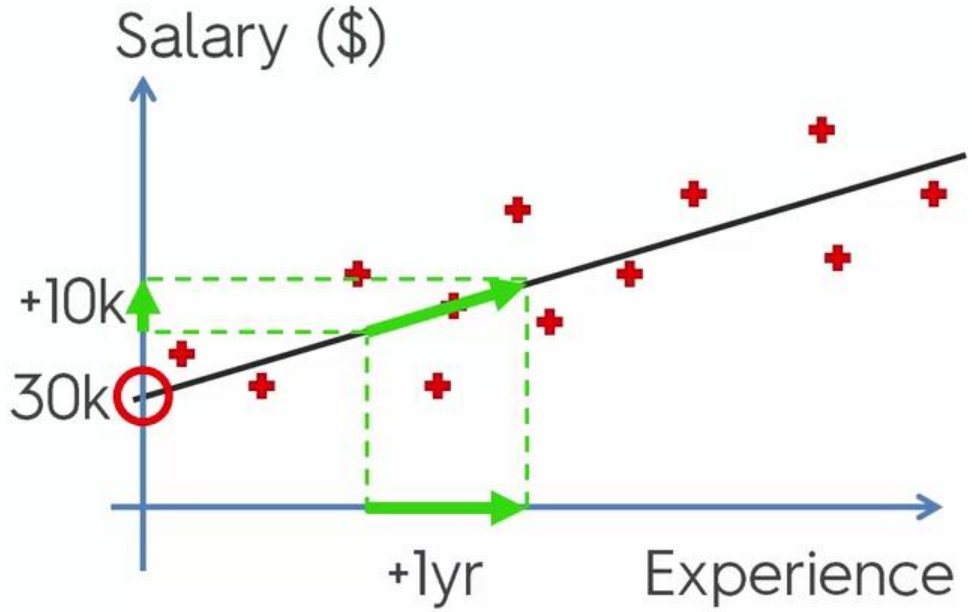


$y = b_0 + b_1 * x$

Salary = b_0 + b_1 *Experience

Regressions

Simple Linear Regression:



$y = b_0 + b_1 * x$

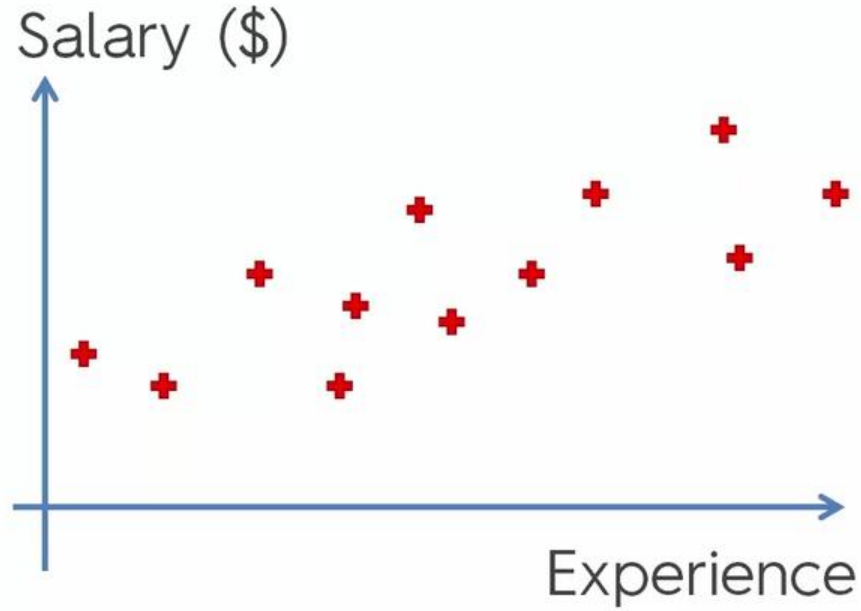
Salary = b_0 + b_1 *Experience

Regressions

Simple Linear Regression:

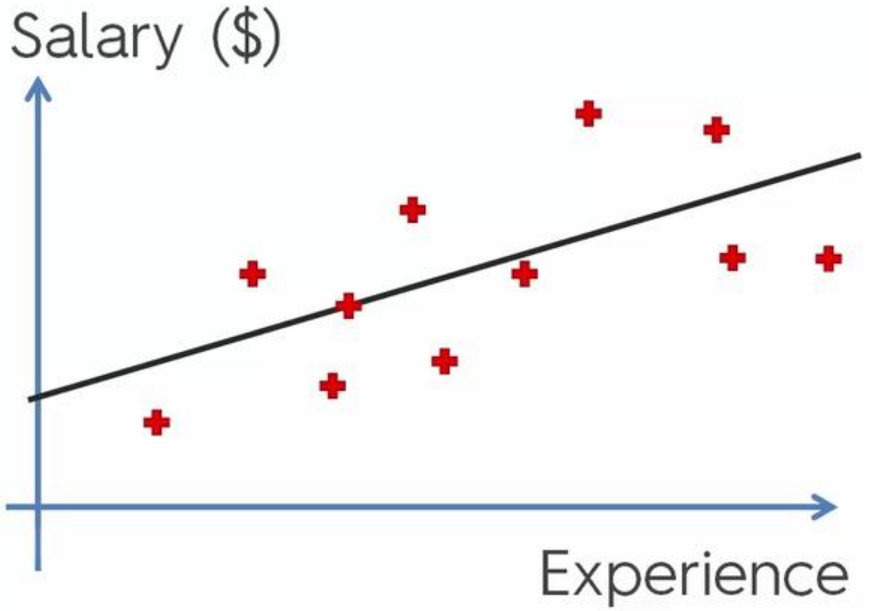
Regressions

Simple Linear Regression:

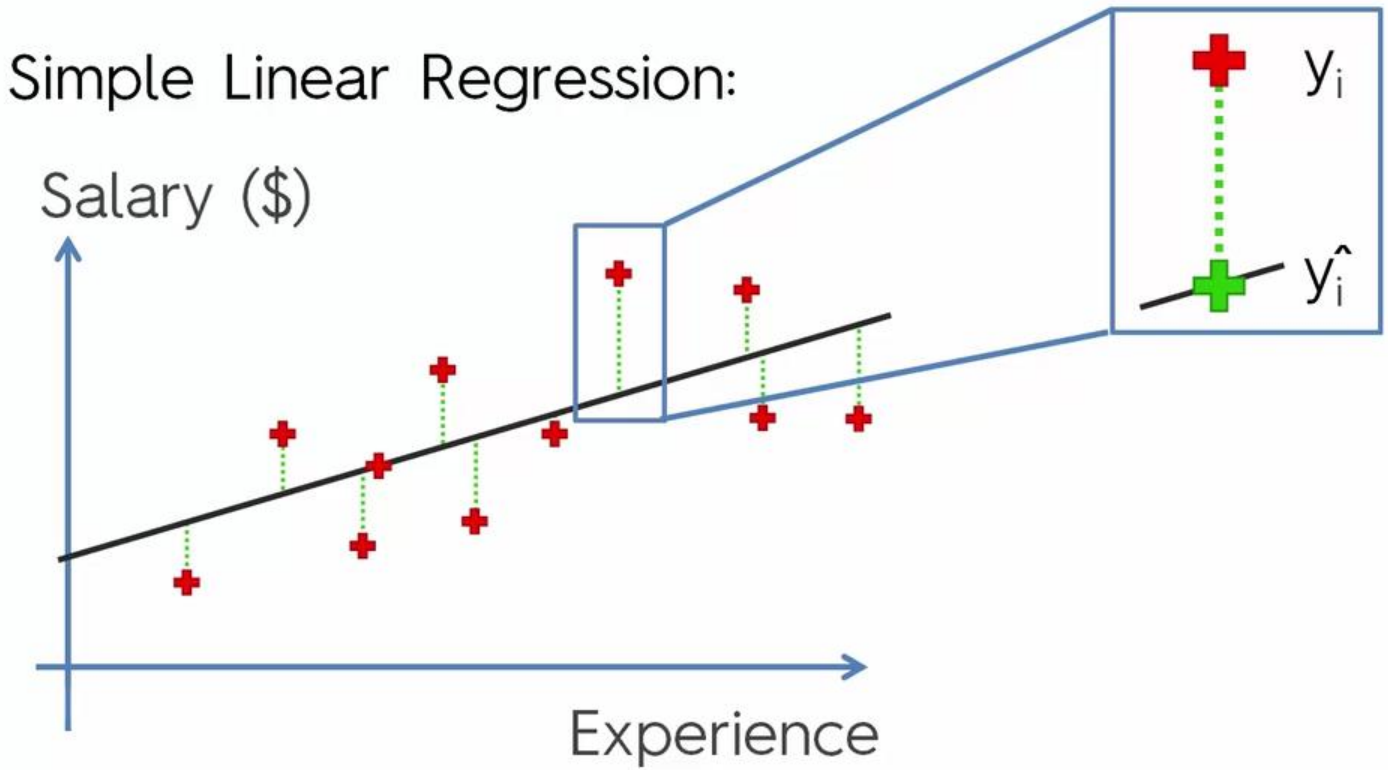


Ordinary Least Squares

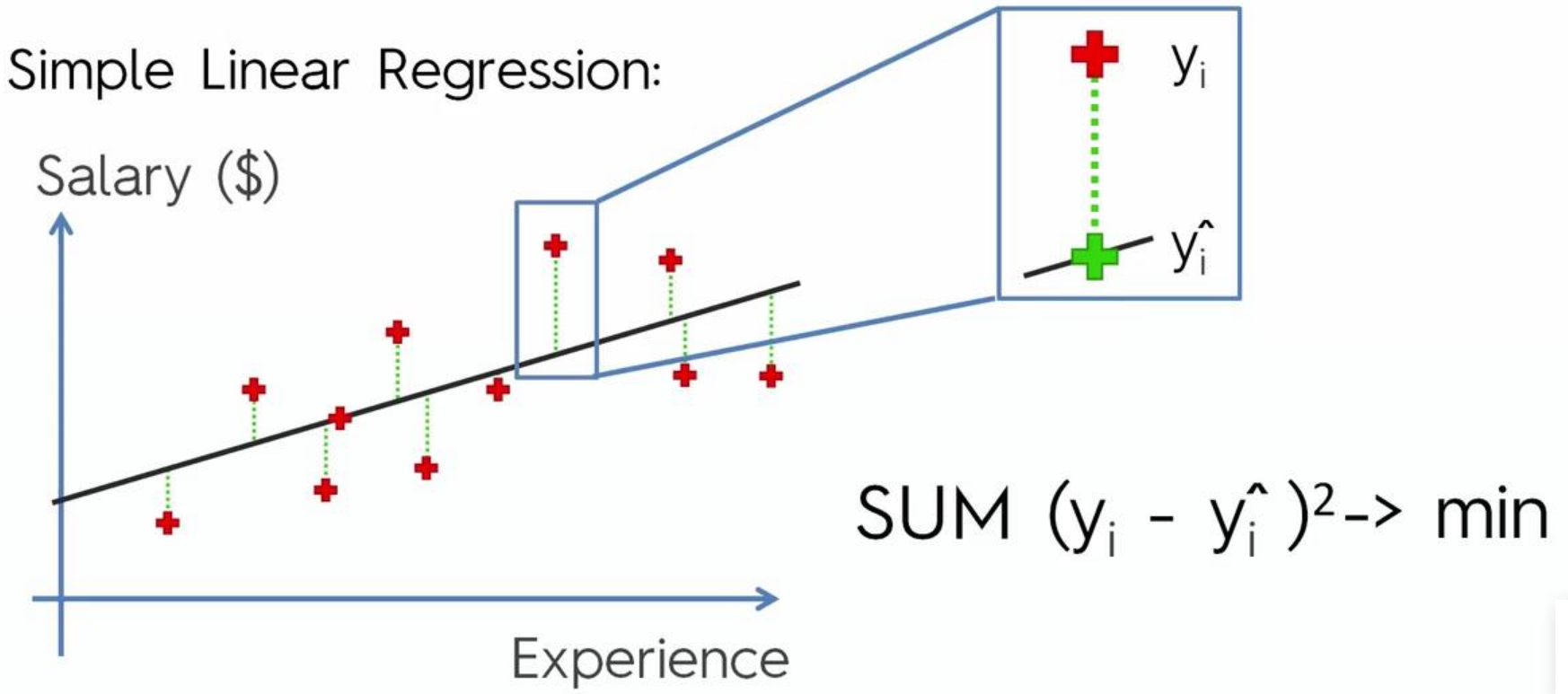
Simple Linear Regression:



Ordinary Least Squares

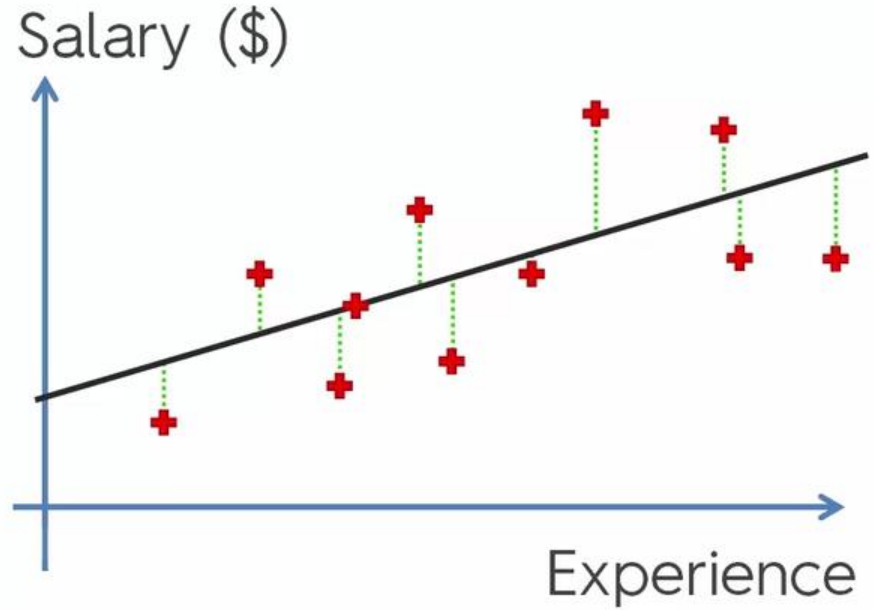


R Squared


Simple Linear Regression:
Salary ($)
Experience
y_i
$\hat{y_i}$
SUM $(y_i - \hat{y_i})^2$ -> min

R Squared

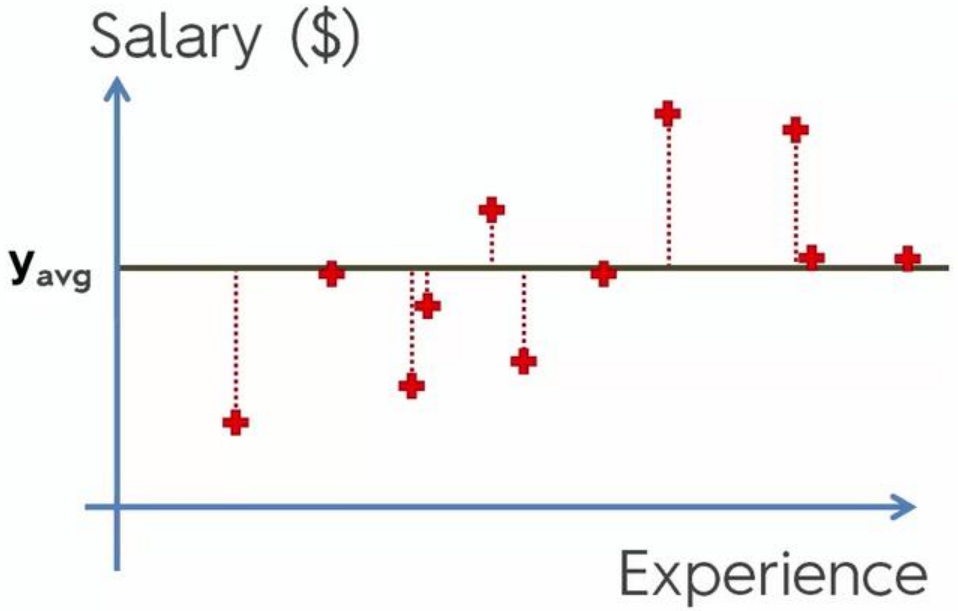
Simple Linear Regression:



$$SS_{res} = SUM\ (y_i - \hat{y}_i)^2$$

R Squared

Simple Linear Regression:

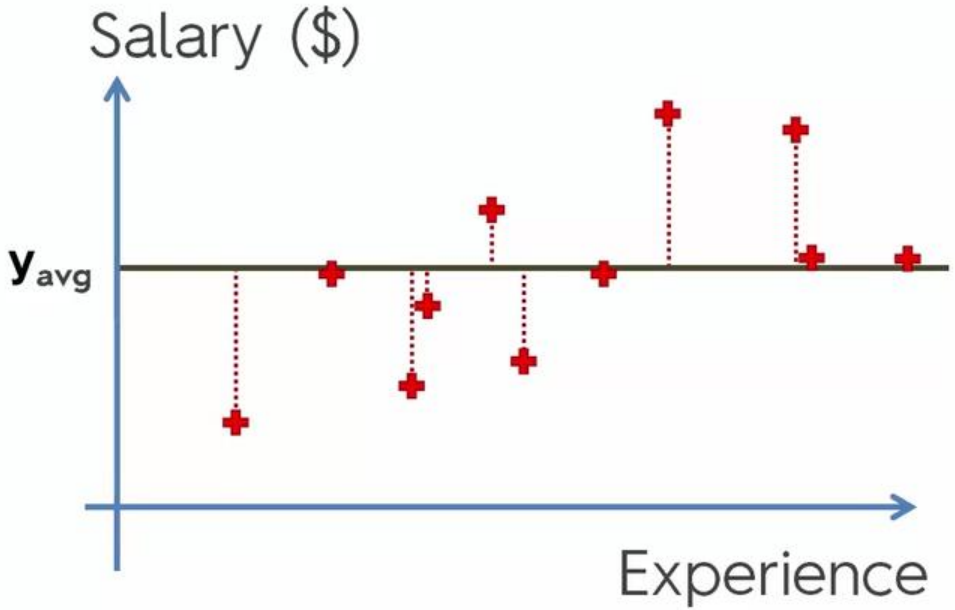


$$SS_{res} = SUM\ (y_i - \hat{y_i})^2$$

$$SS_{tot} = SUM\ (y_i - y_{avg})^2$$

R Squared

Simple Linear Regression:

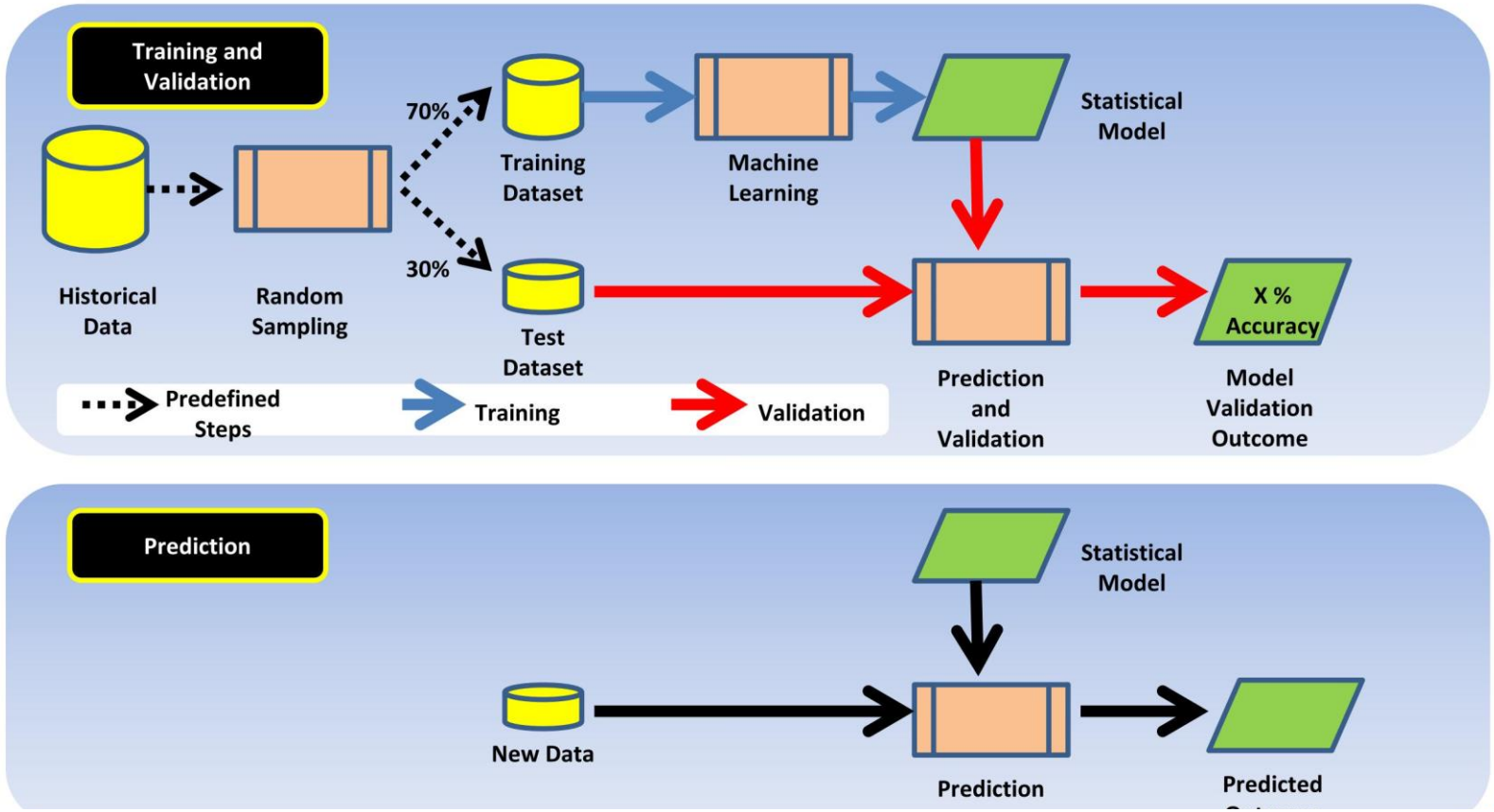


$$SS_{res} = SUM\ (y_i - \hat{y_i})^2$$

$$SS_{tot} = SUM\ (y_i - y_{avg})^2$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Supervised Learning- Process Flow



Dummy Variables

Profit	R&D Spend	Admin	Marketing	State
192,261.83	165,349.20	136,897.80	471,784.10	New York
191,792.06	162,597.70	151,377.59	443,898.53	California
191,050.39	153,441.51	101,145.55	407,934.54	California
182,901.99	144,372.41	118,671.85	383,199.62	New York
166,187.94	142,107.34	91,391.77	366,168.42	California

Dummy Variables

Profit	R&D Spend	Admin	Marketing	State
192,261.83	165,349.20	136,897.80	471,784.10	New York
191,792.06	162,597.70	151,377.59	443,898.53	California
191,050.39	153,441.51	101,145.55	407,934.54	California
182,901.99	144,372.41	118,671.85	383,199.62	New York
166,187.94	142,107.34	91,391.77	366,168.42	California

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + b_3 * x_3 + ???$$

Dummy Variables

Profit	R&D Spend	Admin	Marketing	State	New York	California
192,261.83	165,349.20	136,897.80	471,784.10	New York		
191,792.06	162,597.70	151,377.59	443,898.53	California		
191,050.39	153,441.51	101,145.55	407,934.54	California		
182,901.99	144,372.41	118,671.85	383,199.62	New York		
166,187.94	142,107.34	91,391.77	366,168.42	California		

$$y = b_0 + b_1*x_1 + b_2*x_2 + b_3*x_3 + ???$$

Dummy Variables

Profit	R&D Spend	Admin	Marketing	State	New York	California
192,261.83	165,349.20	136,897.80	471,784.10	New York	1	
191,792.06	162,597.70	151,377.59	443,898.53	California	0	
191,050.39	153,441.51	101,145.55	407,934.54	California	0	
182,901.99	144,372.41	118,671.85	383,199.62	New York	1	
166,187.94	142,107.34	91,391.77	366,168.42	California	0	

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + b_3 * x_3 + ???$$

Dummy Variables

Profit	R&D Spend	Admin	Marketing	State	New York	California
192,261.83	165,349.20	136,897.80	471,784.10	New York	1	0
191,792.06	162,597.70	151,377.59	443,898.53	California	0	1
191,050.39	153,441.51	101,145.55	407,934.54	California	0	1
182,901.99	144,372.41	118,671.85	383,199.62	New York	1	0
166,187.94	142,107.34	91,391.77	366,168.42	California	0	1

$$y = b_0 + b_1*x_1 + b_2*x_2 + b_3*x_3 + ???$$

Dummy Variables

Profit	R&D Spend	Admin	Marketing	State	Dummy Variables: New York	Dummy Variables: California
192,261.83	165,349.20	136,897.80	471,784.10	New York	1	0
191,792.06	162,597.70	151,377.59	443,898.53	California	0	1
191,050.39	153,441.51	101,145.55	407,934.54	California	0	1
182,901.99	144,372.41	118,671.85	383,199.62	New York	1	0
166,187.94	142,107.34	91,391.77	366,168.42	California	0	1

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + b_3 * x_3 + ???$$

Dummy Variables

Profit	R&D Spend	Admin	Marketing	State
192,261.83	165,349.20	136,897.80	471,784.10	New York
191,792.06	162,597.70	151,377.59	443,898.53	California
191,050.39	153,441.51	101,145.55	407,934.54	California
182,901.99	144,372.41	118,671.85	383,199.62	New York
166,187.94	142,107.34	91,391.77	366,168.42	California

Dummy Variables

New York	California
1	0
0	1
0	1
1	0
0	1

$$y = b_0 + b_1*x_1 + b_2*x_2 + b_3*x_3 + b_4*D_1$$

Dummy Variable Trap

Profit	R&D Spend	Admin	Marketing	State
192,261.83	165,349.20	136,897.80	471,784.10	New York
191,792.06	162,597.70	151,377.59	443,898.53	California
191,050.39	153,441.51	101,145.55	407,934.54	California
182,901.99	144,372.41	118,671.85	383,199.62	New York
166,187.94	142,107.34	91,391.77	366,168.42	California

Dummy Variables

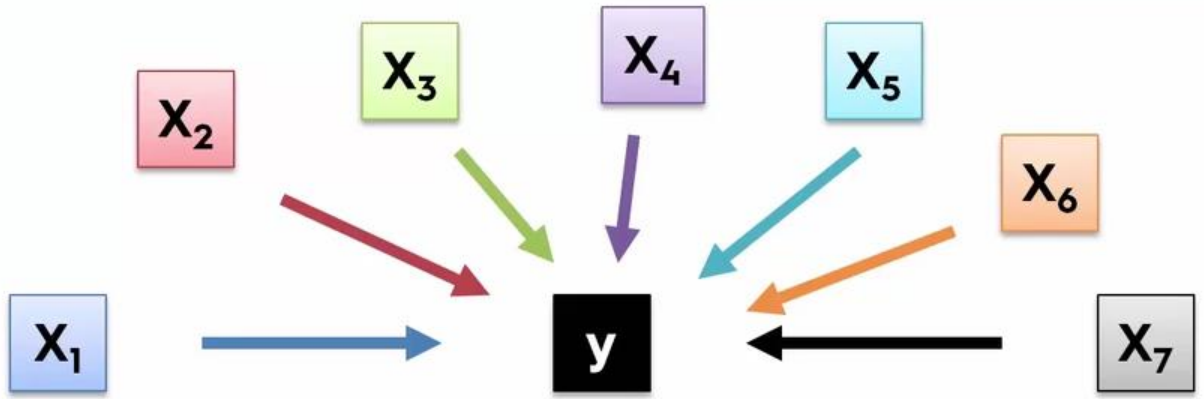
New York	California
1	0
0	1
0	1
1	0
0	1

$$y = b_0 + b_1*x_1 + b_2*x_2 + b_3*x_3 + b_4*D_1 + b_5*D_2$$

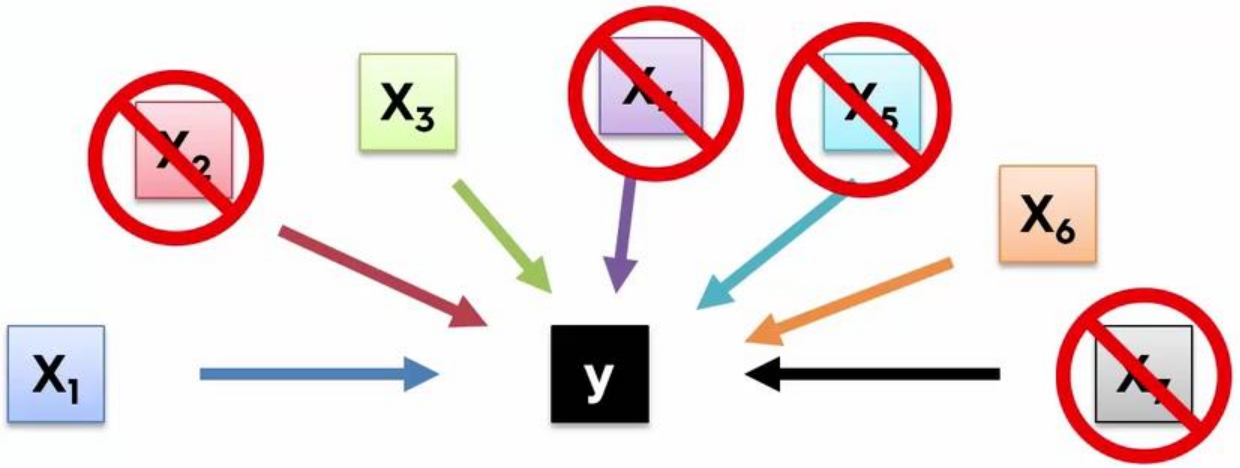
Always omit one dummy variable

Building A Model

Building A Model



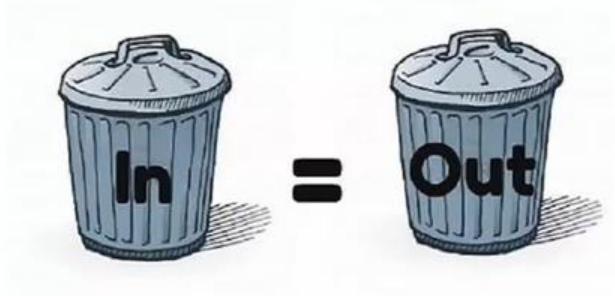
Building A Model



Why?

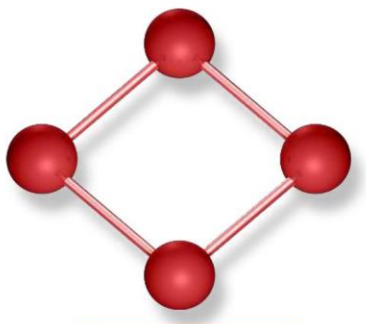
Building A Model

1)



2)

What is Clustering?



Organizing data into *clusters* such that there is:

- ✓ High intra-cluster similarity
- ✓ Low inter-cluster similarity
- ✓ Informally, finding natural groupings among objects.

Why do we want to do it??

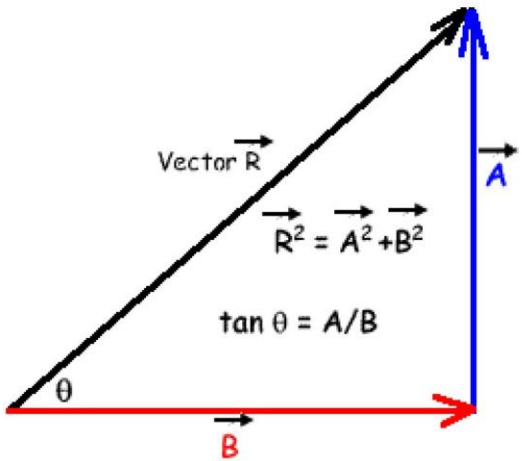
Why Clustering?

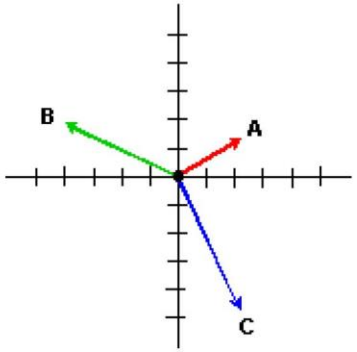
- ✓ Organizing data into clusters shows internal structure of the data
  Ex. Clusty and clustering genes
- ✓ Sometimes the partitioning is the goal
  Ex. Market segmentation
- ✓ Prepare for other AI techniques
  Ex. Summarize news (cluster and then find centroid)
- ✓ Techniques for clustering is useful in knowledge
- ✓ Discovery in data
  Ex. Underlying rules, reoccurring patterns, topics, etc.

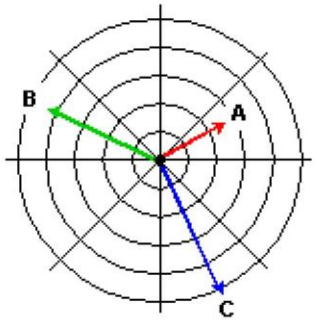
Vector

A **vector** is a quantity or phenomenon that has two independent properties: magnitude and direction.

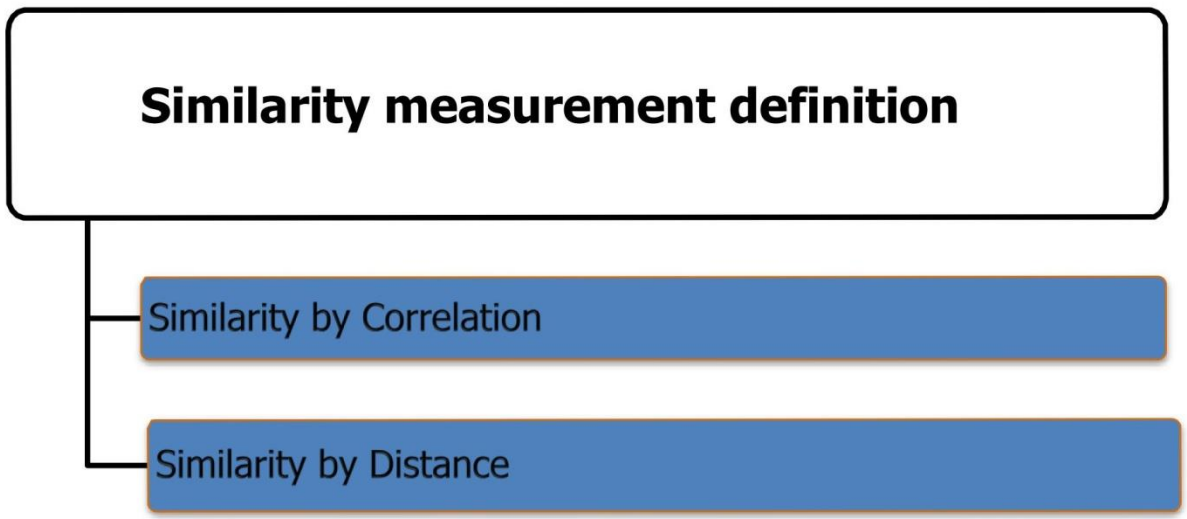
The term also denotes the mathematical or geometrical representation of such a quantity.



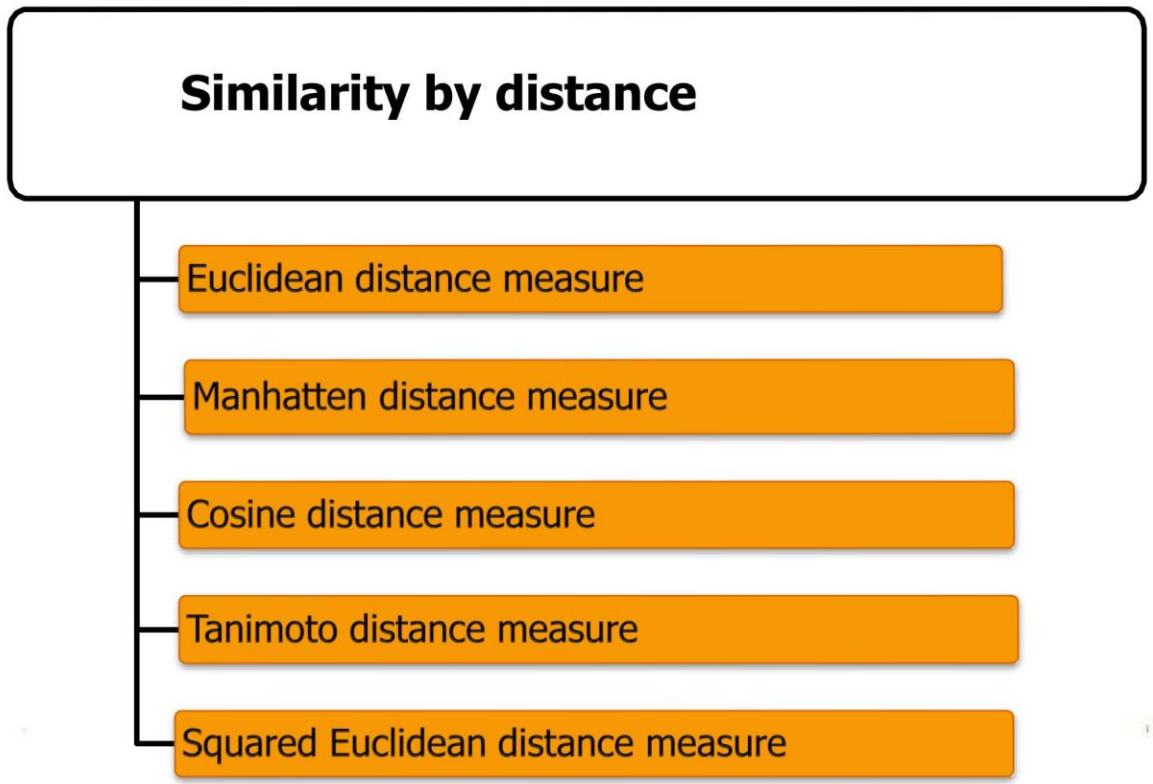




Similarity Measurement



Distance measures



Euclidean distance measure

Mathematically, Euclidean distance between two n-dimensional vectors

(a1, a2, ... , an) and (b1,b2,...,bn) is:

$$d = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_n - b_n)^2}$$

Manhatten distance measure

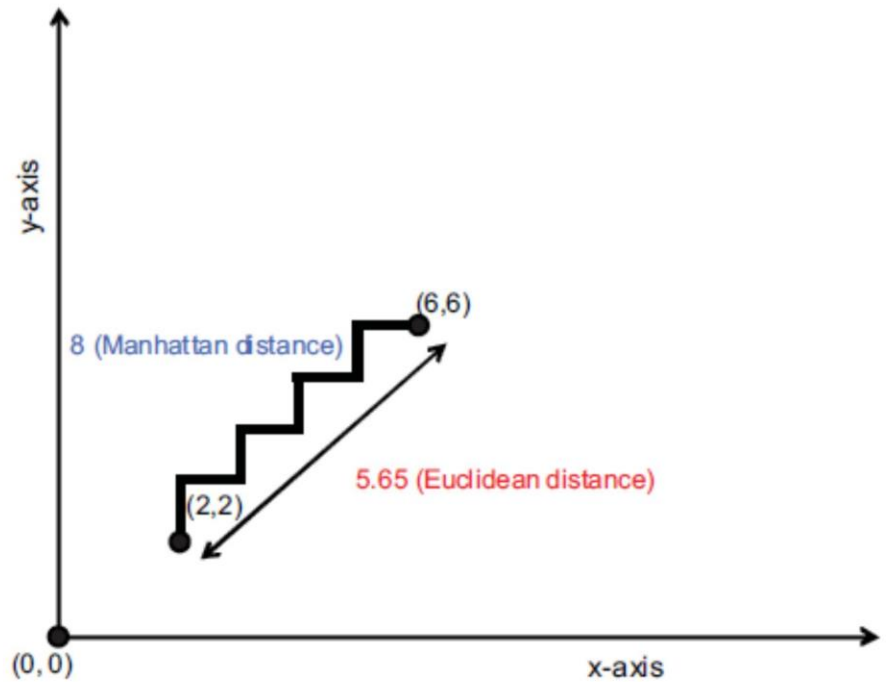
Mathematically, the Manhattan distance between two n-dimensional vectors

(a1, a2, ... , an) and (b1, b2, ... , bn) is

$$d = |a1 - b1| + |a2 - b2| + ... + |an - bn|$$

Difference between Euclidean and Manhattan

From this image we can say that, The Euclidean distance measure gives 5.65 as the distance between (2, 2) and (6, 6) whereas the Manhattan distance is 8.0



Cosine distance measure

The formula for the cosine distance between *n*-dimensional vectors (a1, a2, ... , an) and (b1, b2, ...,bn) is

$$d = 1 - \frac{(a_1 b_1 + a_2 b_2 + \ldots + a_n b_n)}{(\sqrt{(a_1^2 + a_2^2 + \ldots + a_n^2)})\sqrt{(b_1^2 + b_2^2 + \ldots + b_n^2)})}$$

Tanimoto distance measure

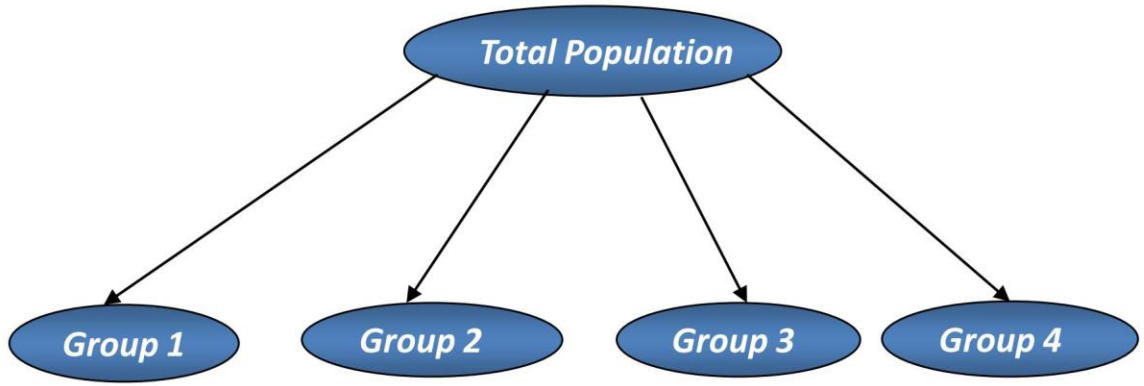
The formula for the Tanimoto distance between two *n*-dimensional vectors (a1, a2, ... , an) and (b1, b2, ... , bn) is

$$d = 1 - \frac{(a_1 b_1 + a_2 b_2 + \ldots + a_n b_n)}{\sqrt{(a_1^2 + a_2^2 + \ldots + a_n^2)} + \sqrt{(b_1^2 + b_2^2 + \ldots + b_n^2)} - (a_1 b_1 + a_2 b_2 + \ldots + a_n b_n)}$$

K-Means clustering

K-Means clustering

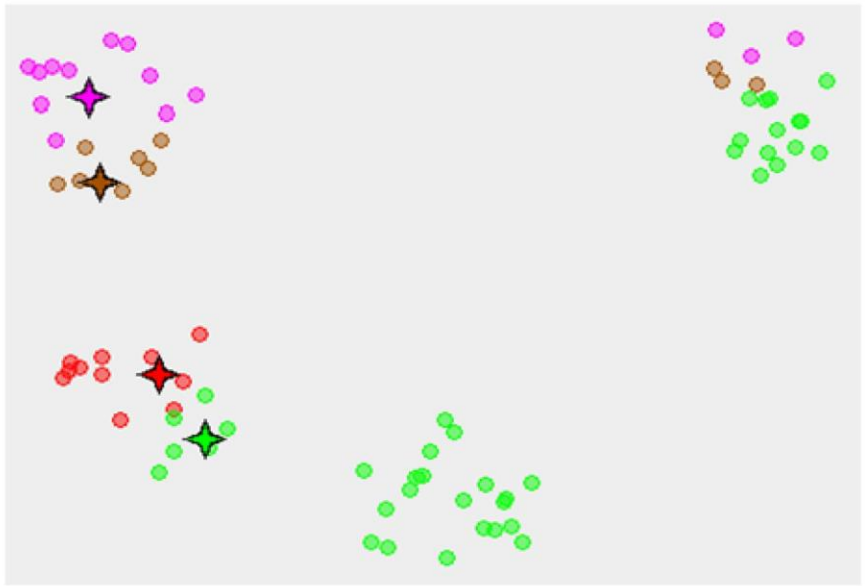
- ✓ The process by which objects are classified into a number of groups so **that they are as much dissimilar as possible from one group to another group, but as much similar as possible within each group.**
- ✓ In other words Cluster analysis means dividing the whole population into groups which are distinct between themselves but internally similar.



- ✓ The objects in group 1 should be as similar as possible.
- ✓ But there should be much difference between an object in group 1 and group 2.
- ✓ The attributes of the objects are allowed to determine which objects should be grouped together.

K-Means clustering steps

1. k initial "means" (in this case k=3) are randomly generated within the data domain.
2. k clusters are created by associating every observation with the nearest mean.
3. The centroid of each of the k clusters becomes the new mean.
4. Steps 2 and 3 are repeated until convergence has been reached.



Step by Step pictorial representation of K-Means clustering

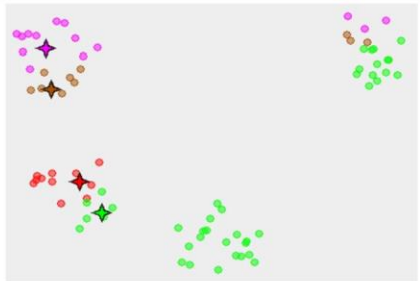


figure-1

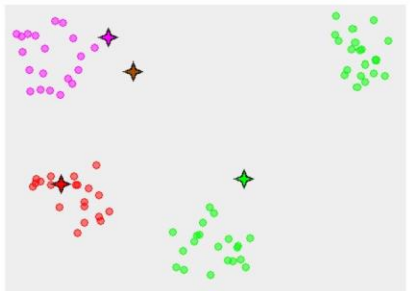


figure-2

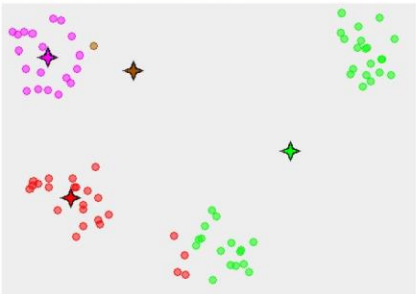


figure-3

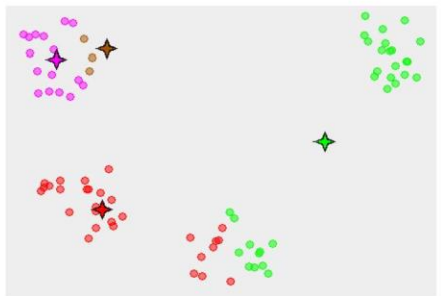


figure-4

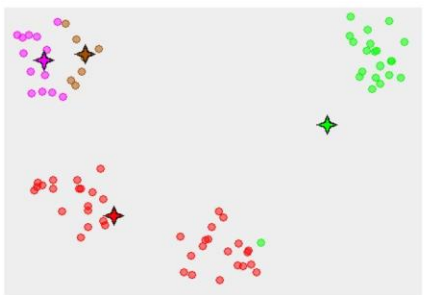


figure-5

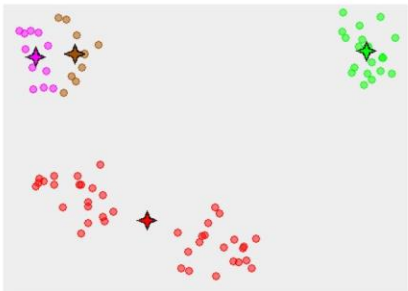


figure-6

- ✓ The small circles are the data points, the four ray stars are the centroids (means).
- ✓ The initial configuration is on the figure-1.
- ✓ The algorithm converges after five iterations presented on the figures, from figure-2 to figure-6.

Association Rule Mining

Association Rule Mining

- In data mining, **association rule learning** is a popular and well researched method for discovering interesting relations between variables in large databases.
- It is intended to identify strong rules discovered in databases using different measures of interests.
- The rule found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy hamburger meat.
- Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements.

Association Rule Mining

SAMPLE INPUT DATA

transaction_id	items
1	citrus fruit
1	semi-finished bread
1	margarine
1	ready soups
2	tropical fruit
2	yogurt
2	coffee
3	whole milk
4	pip fruit
4	yogurt
4	cream cheese
4	meat spreads
5	other vegetables
5	whole milk

Association Rule Mining

	lhs	rhs	support	confidence	lift
89	Hard cheese	Whole milk	0.01006609	0.41078838	1.6076815
90	Whole milk	Hard cheese	0.01006609	0.03939515	1.6076815
91	Butter milk	Other vegetables	0.01037112	0.37090909	1.9169159
92	Other vegetables	Butter milk	0.01037112	0.05359958	1.9169159
93	Butter milk	Whole milk	0.01159126	0.41454545	1.6223854
94	Whole milk	Butter milk	0.01159126	0.04536411	1.6223854
95	ham	Whole milk	0.01148958	0.44140625	1.7275091
96	Whole milk	ham	0.01148958	0.04496618	1.7275091
97	Sliced cheese	Whole milk	0.01077783	0.43983402	1.7213560
98	Whole milk	Sliced cheese	0.01077783	0.04218066	1.7213560
99	oil	Whole milk	0.01128622	0.40217391	1.5739675
100	Whole milk	Oil	0.01128622	0.04417031	1.5739675
101	onions	Other vegetables	0.01423488	0.45901639	2.3722681
102	Other vegetables	Onions	0.01423488	0.07356805	2.3722681
103	onions	Whole milk	0.01209964	0.39016393	1.5269647
104	Whole milk	Onions	0.01209964	0.04735376	1.5269647
105	berries	yogurt	0.01057448	0.31804281	2.2798477

Association Rule Mining-Single Cardinality

S No.	Rules	Support	Confidence	Lift
1	{Strawberry Blonde} => {Canterville Ghost}	6.91%	35.91%	1.838296285
2	{Canterville Ghost} => {Strawberry Blonde}	6.91%	35.38%	1.838296285
3	{Doc Savage: The Man of Bronze} => {Green Slime}	8.28%	38.98%	1.791373861
4	{Green Slime} => {Doc Savage: The Man of Bronze}	8.28%	38.06%	1.791373861
5	{Green Slime} => {She}	8.22%	37.80%	1.769506084
6	{She} => {Green Slime}	8.22%	38.50%	1.769506084
7	{Jack the Ripper (1988)} => {She}	5.94%	35.14%	1.644963145
8	{She} => {Jack the Ripper (1988)}	5.94%	27.81%	1.644963145
9	{Pretty Maids All In A Row} => {Dark of the Sun}	7.37%	34.22%	1.580866863
10	{Dark of the Sun} => {Pretty Maids All In A Row}	7.37%	34.04%	1.580866863
11	{Doc Savage: The Man of Bronze} => {She}	6.97%	32.80%	1.535434995
12	{She} => {Doc Savage: The Man of Bronze}	6.97%	32.62%	1.535434995
13	{Pretty Maids All In A Row} => {Green Slime}	6.85%	31.83%	1.462854278
14	{Green Slime} => {Pretty Maids All In A Row}	6.85%	31.50%	1.462854278
15	{Pretty Maids All In A Row} => {She}	6.62%	30.77%	1.440559441

Sample Interpretation for Rule 1: Those customers buying Strawberry Blonde are usually more prone to also buy Canterville Ghost.

Association Rule Mining-Multiple Cardinalities

S No.	Rules	Support	Confidence	Lift
1	{Green Slime,Jack the Ripper (1988)} => {She}	3.14%	75.34%	3.52739726
2	{Canterville Ghost,Dark of the Sun} => {Strawberry Blonde}	2.57%	66.18%	3.4384273
3	{Jack the Ripper (1988),Strawberry Blonde} => {Canterville Ghost}	2.51%	65.67%	3.362311251
4	{She,Strawberry Blonde} => {Canterville Ghost}	2.51%	63.77%	3.264852954
5	{Canterville Ghost,Pretty Maids All In A Row} => {Strawberry Blonde}	2.57%	62.50%	3.247403561
6	{Dark of the Sun,Doc Savage: The Man of Bronze,She} => {Green Slime}	2.11%	69.81%	3.208389046
7	{Dark of the Sun,Doc Savage: The Man of Bronze,Green Slime} => {She}	2.11%	68.52%	3.207912458
8	{Doc Savage: The Man of Bronze,Pretty Maids All In A Row,She} => {Green Slime}	2.06%	69.23%	3.181708056
9	{Dark of the Sun,Strawberry Blonde} => {Canterville Ghost}	2.57%	60.81%	3.11344239
10	{Pretty Maids All In A Row,Strawberry Blonde} => {Canterville Ghost}	2.57%	60.00%	3.071929825
11	{Doc Savage: The Man of Bronze,Pretty Maids All In A Row} => {Green Slime}	3.26%	66.28%	3.046053836
12	{Doc Savage: The Man of Bronze,Jack the Ripper (1988)} => {She}	2.23%	65.00%	3.043181818
13	{Canterville Ghost,Jack the Ripper (1988)} => {Strawberry Blonde}	2.51%	57.89%	3.008121193
14	{Doc Savage: The Man of Bronze,Green Slime,Pretty Maids All In A Row} => {She}	2.06%	63.16%	2.956937799
15	{Doc Savage: The Man of Bronze,Stranger on the Third Floor} => {Green Slime}	2.57%	64.29%	2.954443195

Sample Interpretation for Rule 1: Those customers who buy 'Green Slime' and 'Jack the Ripper' are generally more prone to buy 'She' also.

Association Rule Mining - Concepts

Constraints on below measures are used to select useful and best rules of all the rules given by R
After analyzing these values for all the rules, best rules for WB have been obtained.

Support

- The support Supp(X)=proportion of transactions in the data set which contain the interest.

Confidence

- The confidence of a rule:
  Conf(x=>y)= Supp(X U Y)/Supp(X)

Lift

- The lift of a rule: Lift(X=>Y)=
  Supp(X U Y)

  (Supp(X) X Supp(Y))

E.g.:- Consider rule: {Jack the Ripper (1988)} => {Strawberry Blonde}
Let Jack the Ripper =X and Strawberry Blonde =Y, Then

Support(X U Y)= No of transactions involving both Jack the Ripper and Strawberry Blonde/ Total no of transactions

Confidence= No of transactions where Strawberry Blonde was also bought when Jack the Ripper was bought/ No of transactions where Jack the Ripper was bought

Lift = Ratio of observed support to the expected support

Association Rule Mining - Concepts

Association rule generation is usually split up into two separate steps:

Step #1:

Minimum support is applied to find all frequent itemsets in a database.

Step #2:

These frequent itemsets and the minimum confidence constraint are used to form rules.

Logistic Regression

Heart Disease Prediction

Heart disease describes a range of conditions that affect your heart. Diseases under the heart disease umbrella include blood vessel diseases, such as coronary artery disease, heart rhythm problems (arrhythmias) and heart defects you're born with (congenital heart defects), among others.

The term “heart disease” is often used interchangeably with the term “cardiovascular disease”. Cardiovascular disease generally refers to conditions that involve narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina) or stroke. Other heart conditions, such as those that affect your heart’s muscle, valves or rhythm, also are considered forms of heart disease.

Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of clinical data analysis. The amount of data in the healthcare industry is huge.

According to a news article, heart disease proves to be the leading cause of death for both women and men.

> About 610,000 people die of heart disease in the United States every year–that's 1 in every 4 deaths.1
>
> Heart disease is the leading cause of death for both men and women. More than half of the deaths due to heart disease in 2009 were in men.1

Coronary Heart Disease(CHD) is the most common type of heart disease, killing over 370,000 people annually.

Every year about 735,000 Americans have a heart attack. Of these, 525,000 are a first heart attack and 210,000 happen in people who have already had a heart attack.

This makes heart disease a major concern to be dealt with. But it is difficult to identify heart disease because of several contributory risk factors such as diabetes, high blood pressure, high cholesterol, abnormal pulse rate, and many other factors. Due to such constraints, scientists have turned towards modern approaches like Data Mining and Machine Learning for predicting the disease.

Machine learning (ML) proves to be effective in assisting in making decisions and predictions from the large quantity of data produced by the healthcare industry.

INTRODUCTION TO PREDICTIVE MODELING

What is Predictive Modeling?

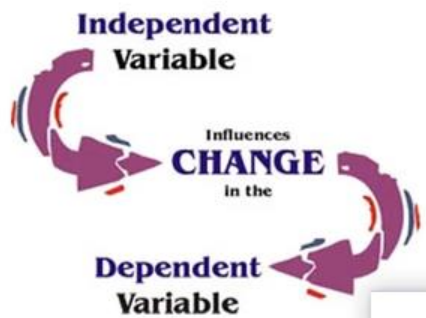
- Building a Statistical model to predict the future behavior
- Predictive Modeling is often referred to as Predictive Analytics
- Popular techniques used for Predictive Modeling:
  - Linear Regression
  - Logistic Regression
  - Classification and Regression Trees
  - Neural Networks
  - Naïve Bayes Classifier

INTRODUCTION TO PREDICTIVE MODELING

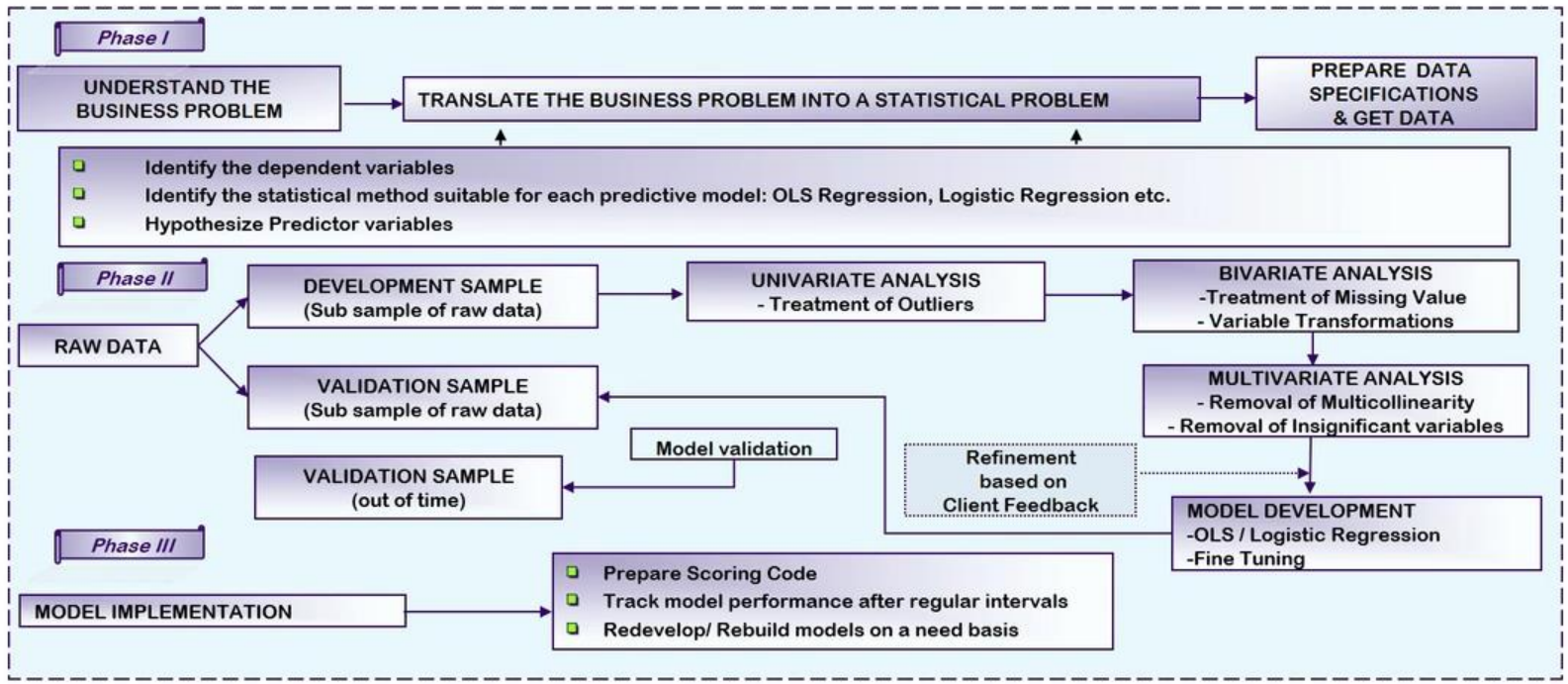
How to build a Predictive Model?

- Predictive Model needs 2 sets of variables –
  - Target, Response or "Dependent" variable (Y)
  - Predictor or "Independent" variables (X_i)
- Estimate a mathematical relationship between the Dependent & Independent variables
- Example
  - Dependent Variable: Purchase the product (yes or no)?
  - Independent Variables: Age, Gender, Salary, Marital Status, Number of Dependents, Savings Account Balance, Mortgage, Other Loans and so on

$$y = ax + b$$



PREDICTIVE MODELING PROCESS



LOGISTIC REGRESSION

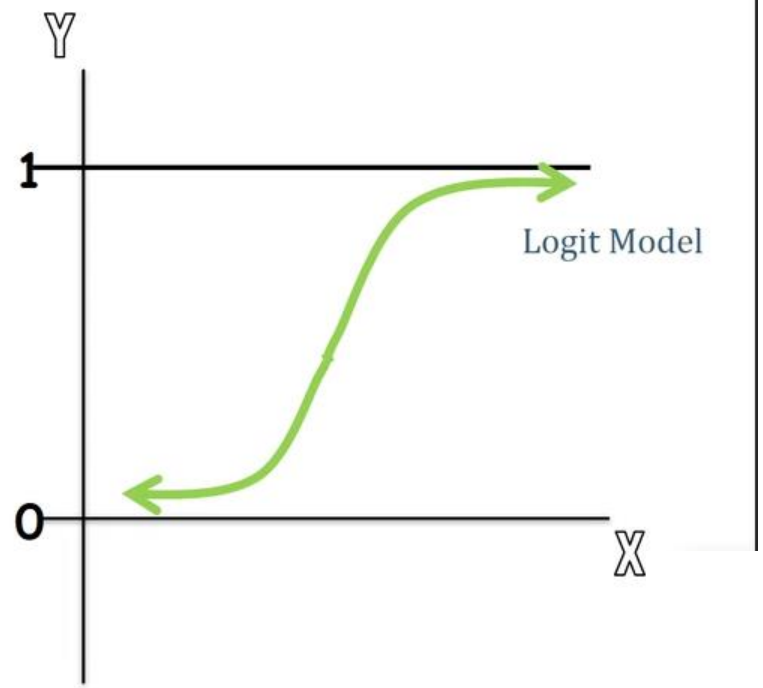
- Logistic Regression predicts the probability of occurrence of an event
- Logistic Regression analyzes the relationship between a dichotomous dependent variable and the independent variables
- Logistic Regression can be used to answer the questions like -
  - What is the probability that the debtor will pay back the loan?
  - What is the probability that the customer will churn?
  - What is the probability that the student will pass the test?
  - What is the probability that the claim is fraud?
- Applications across domains like Finance, Healthcare, Retail, Telecom, Insurance and so on

LOGISTIC REGRESSION - EQUATION

- The equation for Logistic Regression is:

$$\ln[p/(1-p)] = \alpha + \beta X + e$$

  - p is the probability that the event Y occurs
  - p/(1-p) is the "odds ratio"
  - ln[p/(1-p)] is the log odds ratio, or "logit"
- The logistic distribution constrains the estimated probabilities to lie between 0 and 1
- The estimated probability is: $p = 1/[1 + \exp(-\alpha - \beta X)]$



Logistic Regression

Linear Regression:

- **Simple:**

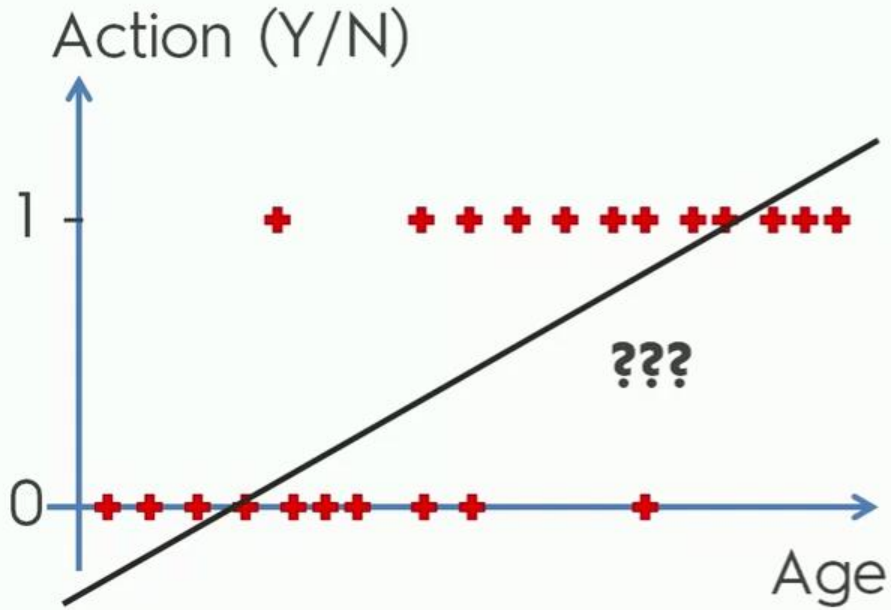
$$y = b_0 + b_1 * x$$

- **Multiple:**

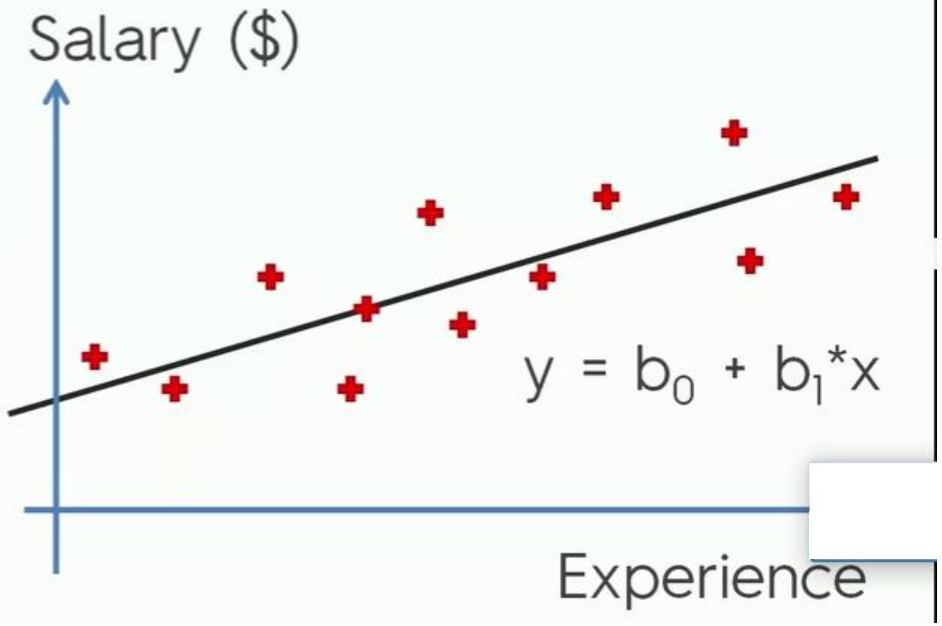
$$y = b_0 + b_1 * x_1 + \ldots + b_n * x_n$$

Logistic Regression

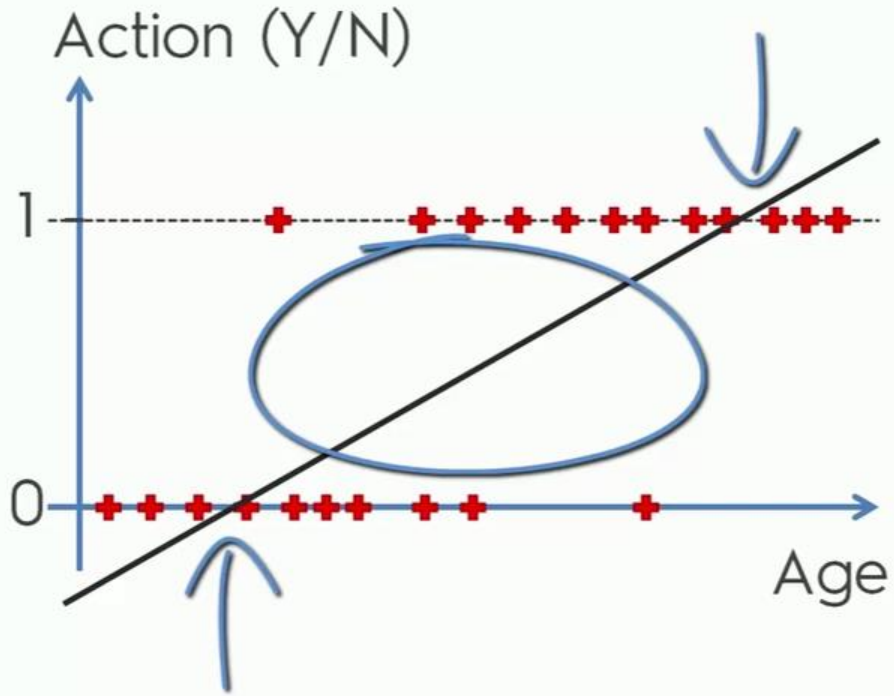
This is new:



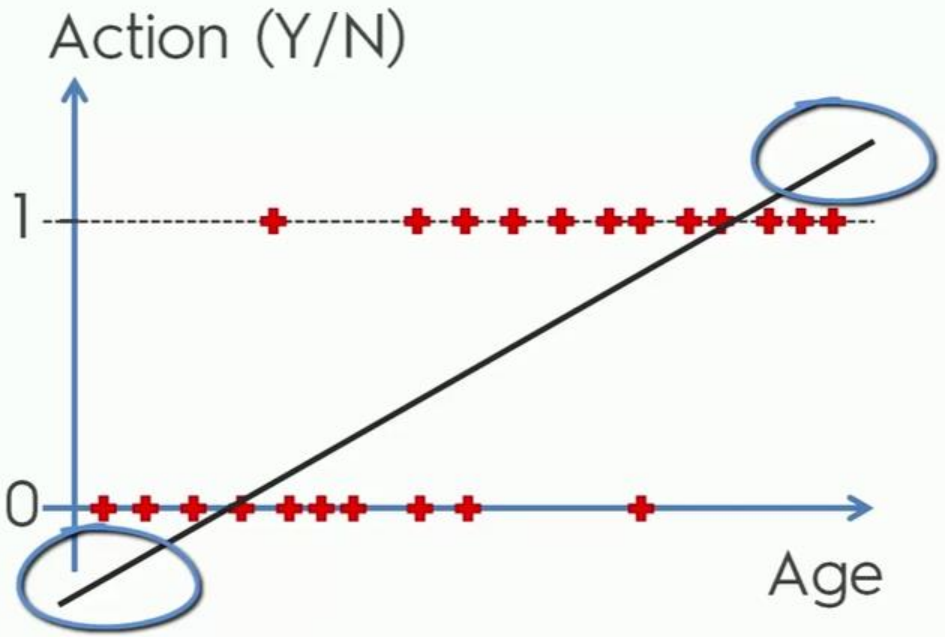
We know this:



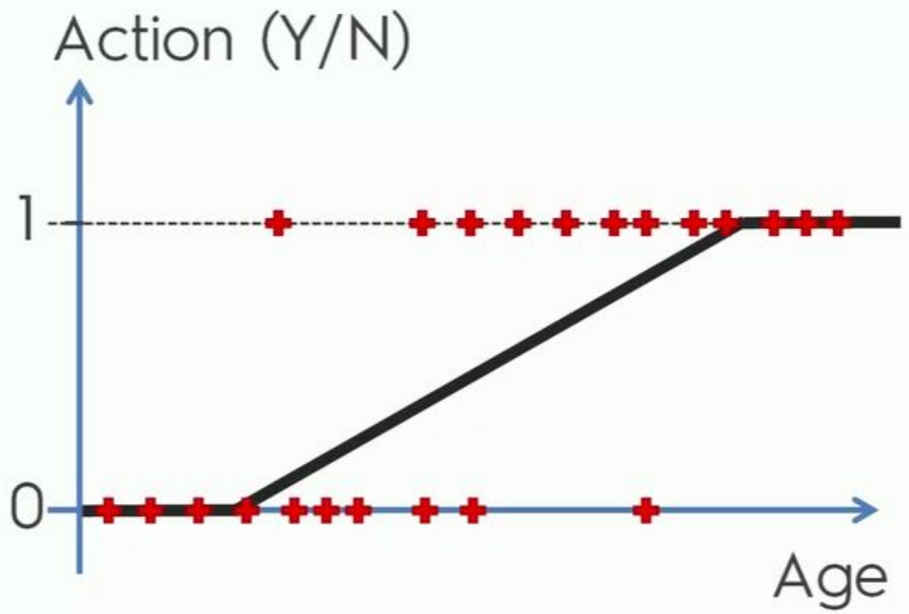
Logistic Regression



Logistic Regression



Logistic Regression



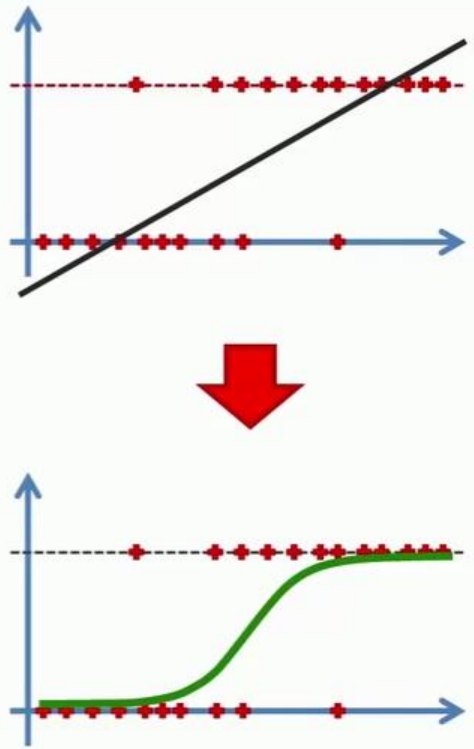
Logistic Regression

$$y = b_0 + b_1 * x$$

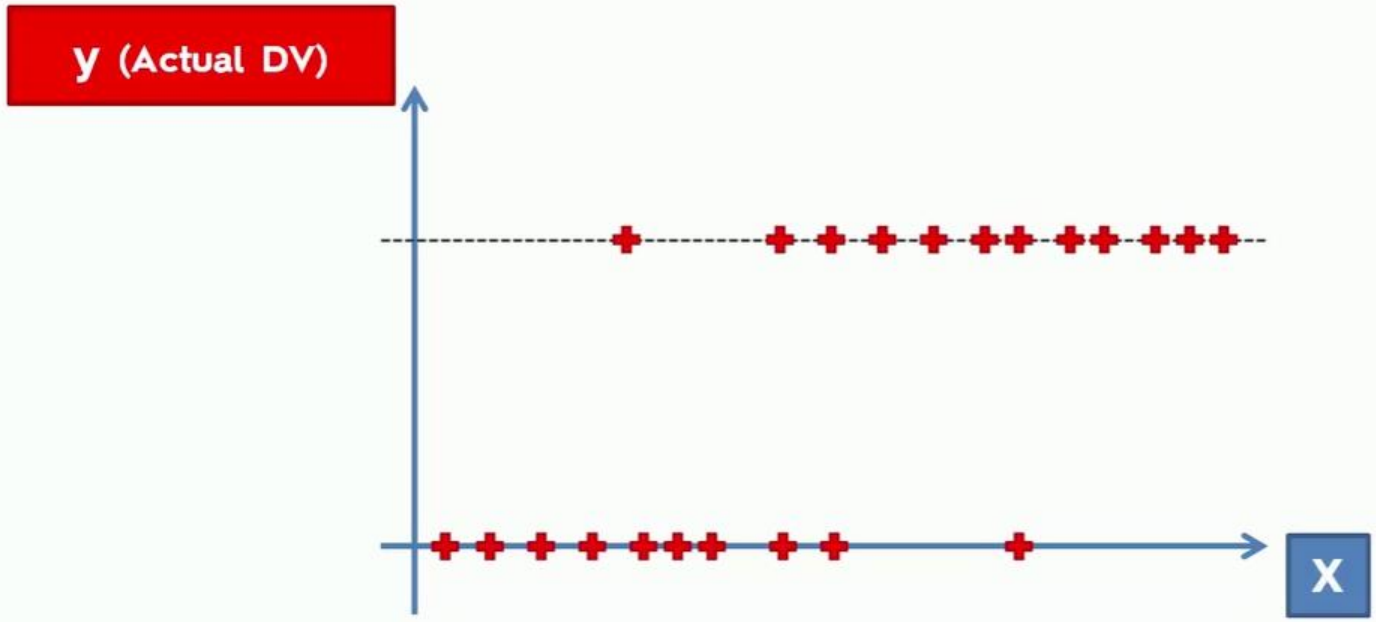
Sigmoid Function

$$p = \frac{1}{1 + e^{-y}}$$

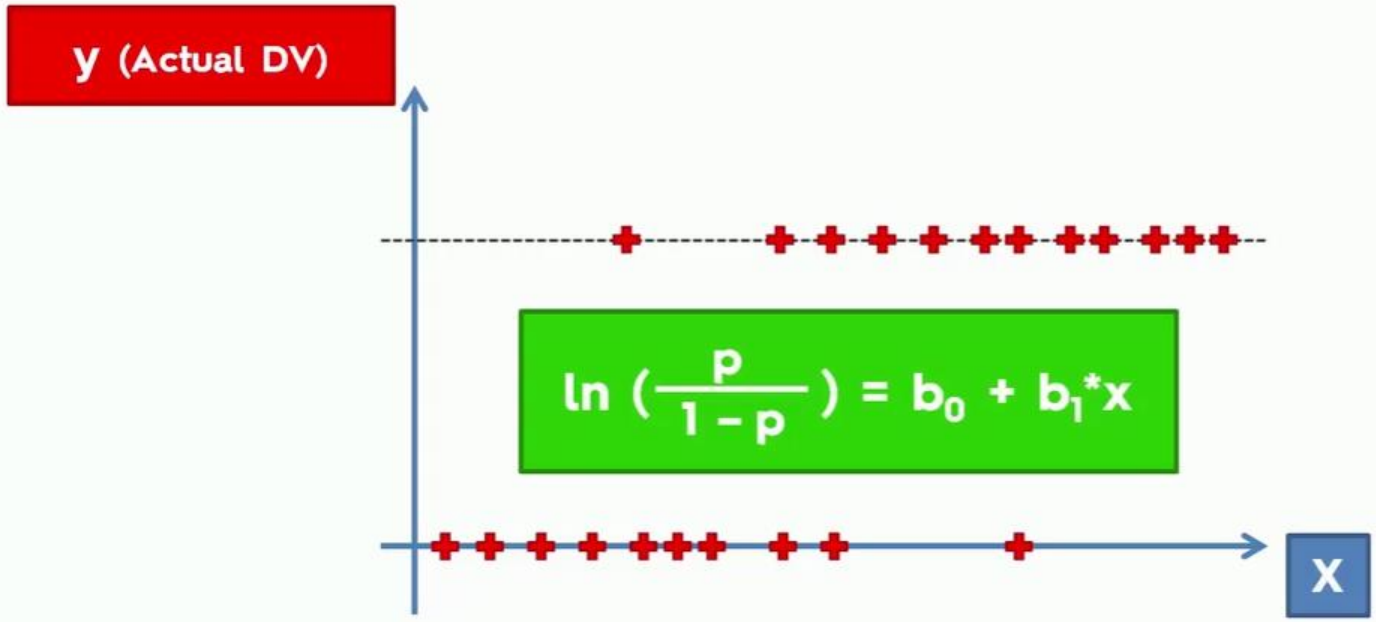
$$\ln\left(\frac{p}{1-p}\right) = b_0 + b_1 * x$$



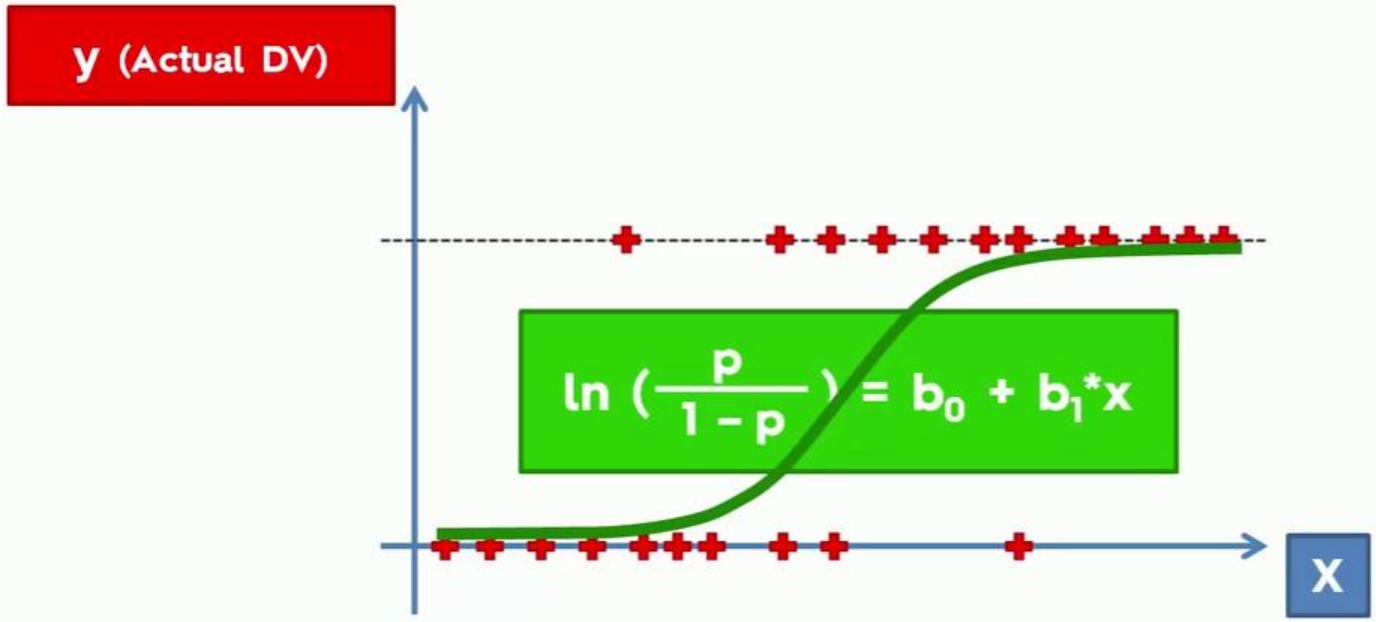
Logistic Regression



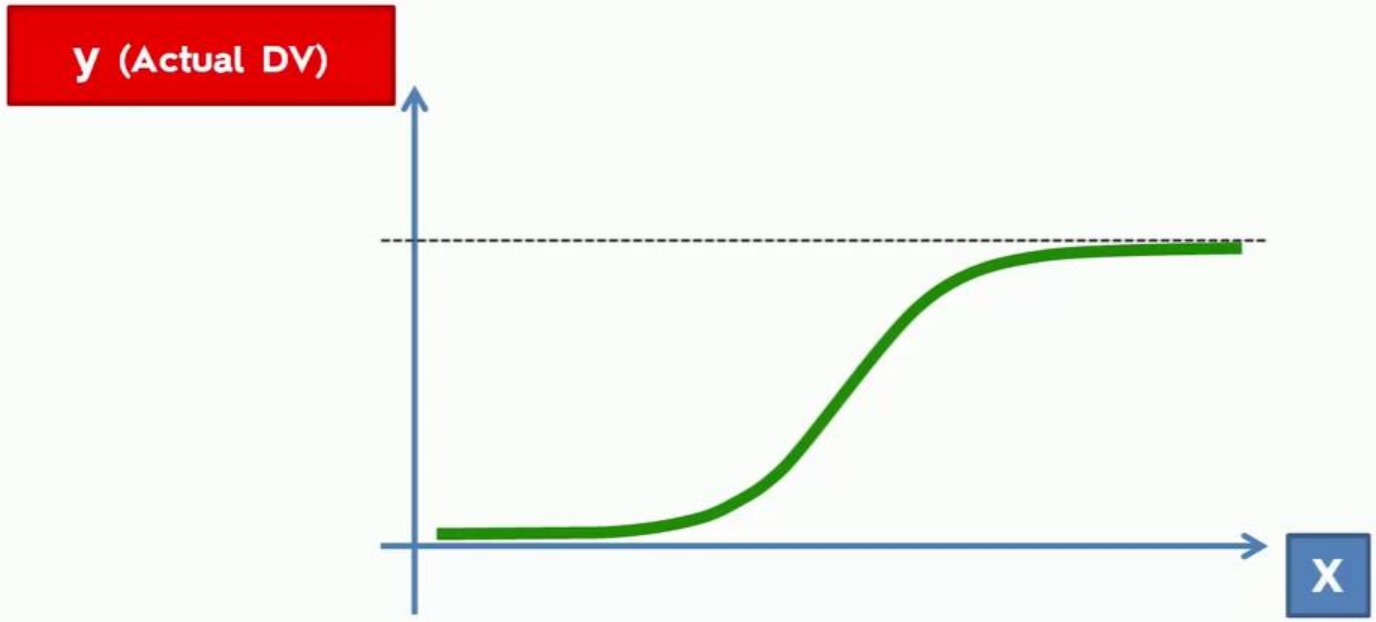
Logistic Regression



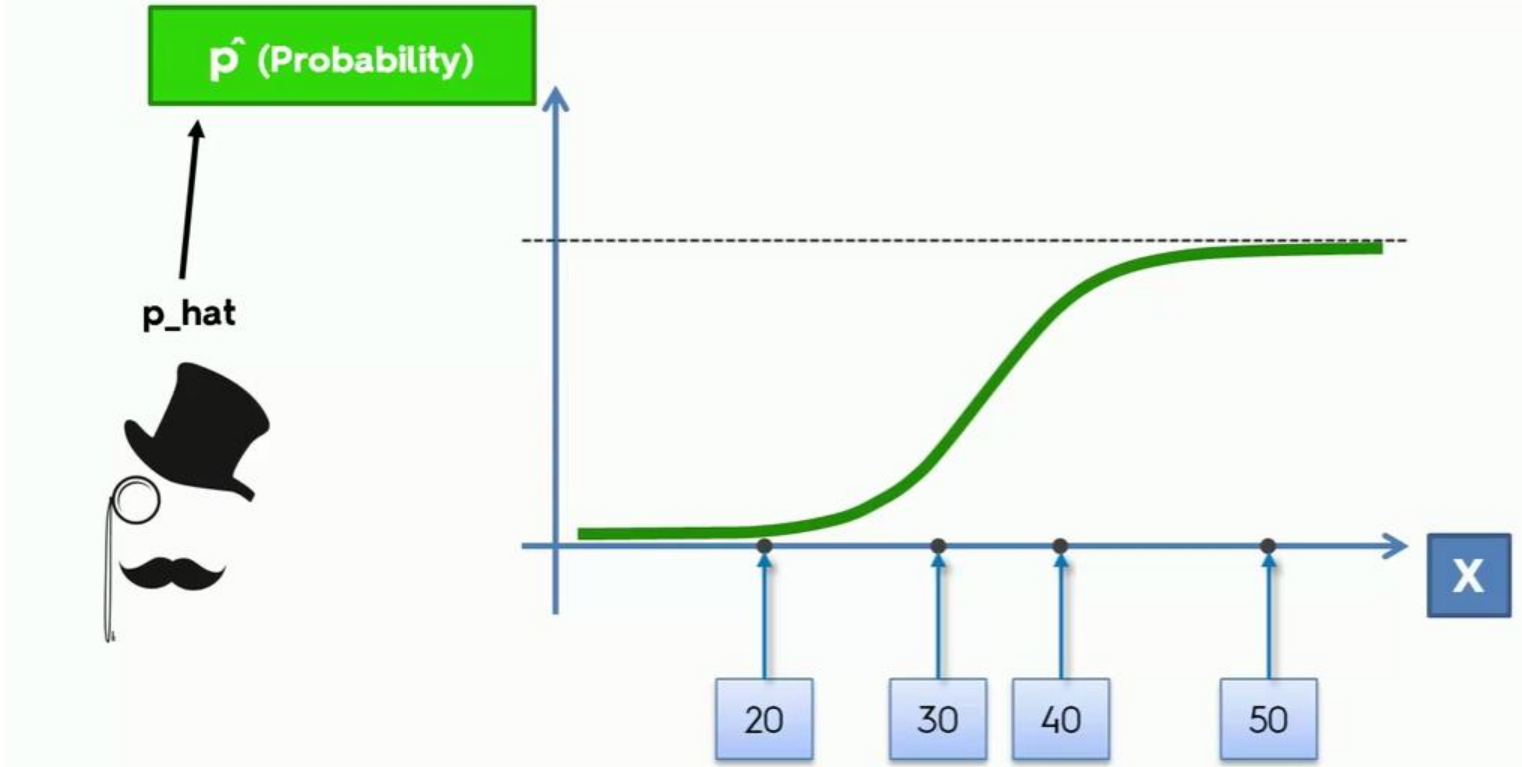
Logistic Regression



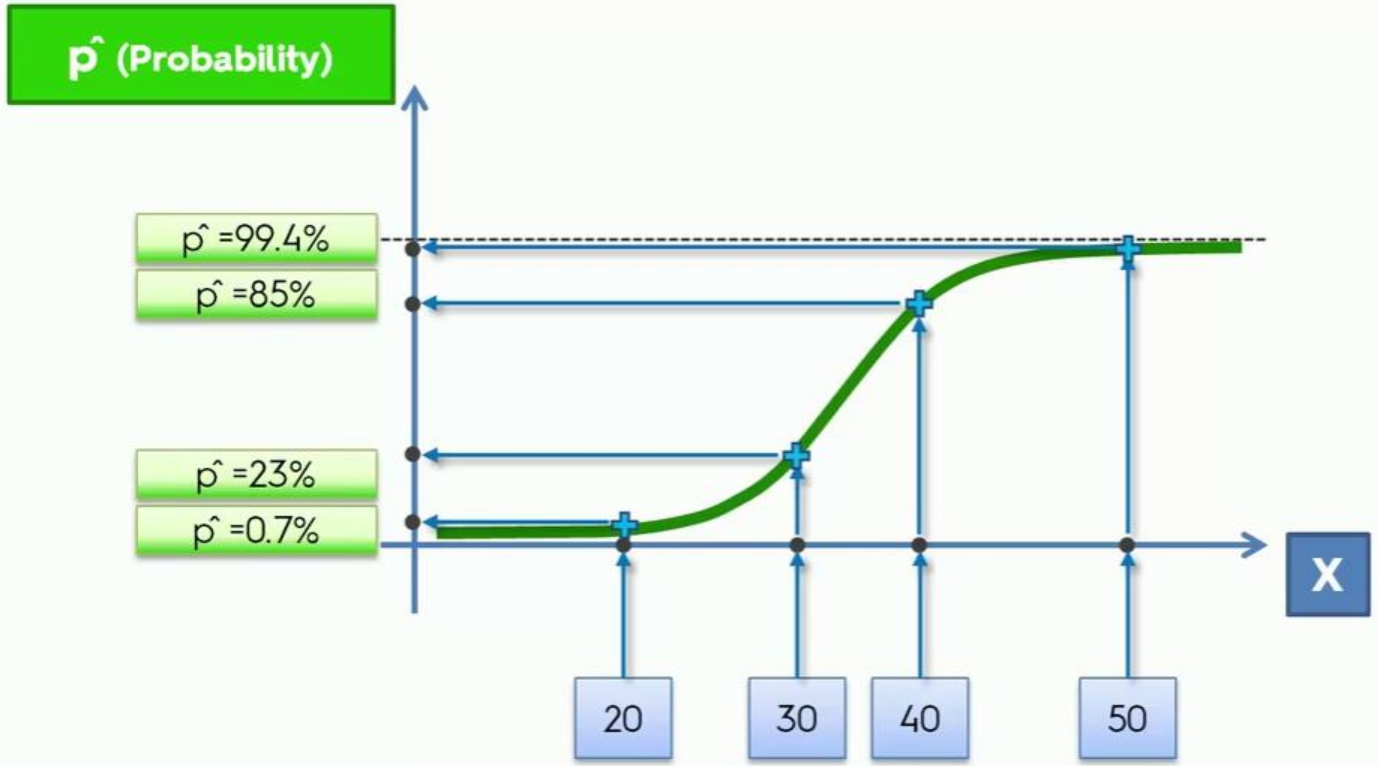
Logistic Regression



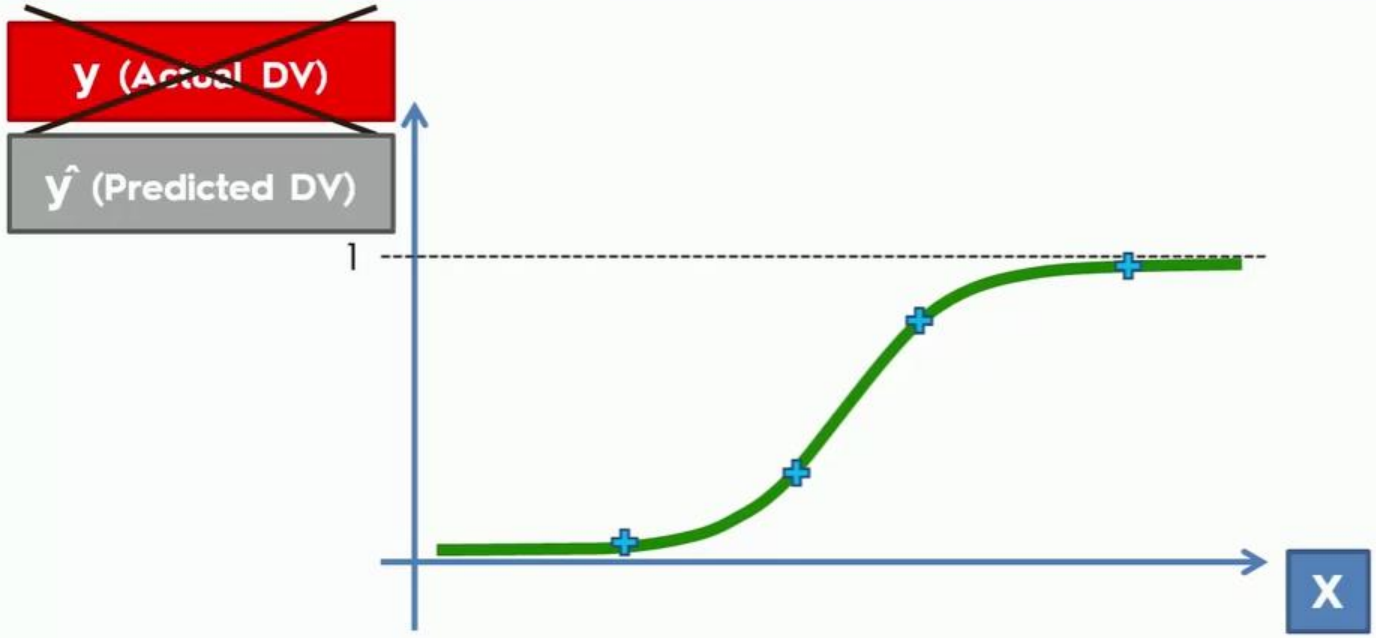
Logistic Regression



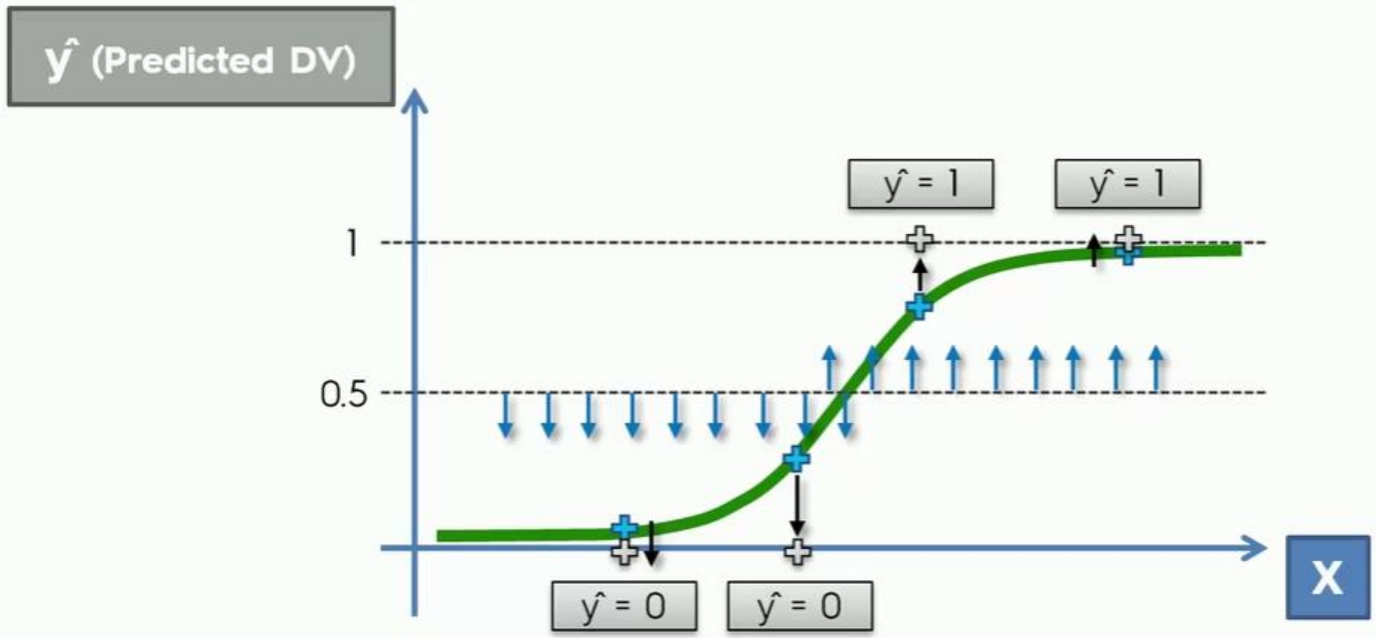
Logistic Regression



Logistic Regression



Logistic Regression



The dataset consists of 303 individuals data. There are 14 columns in the dataset, which are described below.

1. ***Age***: displays the age of the individual.
2. ***Sex***: displays the gender of the individual using the following format :
   1 = male
   0 = female

3. ***Chest-pain type***: displays the type of chest-pain experienced by the individual using the following format :

   1 = typical angina

   2 = atypical angina

   3 = non — anginal pain

   4 = asymptotic

4. ***Resting Blood Pressure***: displays the resting blood pressure value of an individual in mmHg (unit)

5. ***Serum Cholestrol***: displays the serum cholesterol in mg/dl (unit)

6. ***Fasting Blood Sugar***: compares the fasting blood sugar value of an individual with 120mg/dl.
   If fasting blood sugar > 120mg/dl then : 1 (true)
   else : 0 (false)

7. ***Resting ECG*** : displays resting electrocardiographic results
   0 = normal
   1 = having ST-T wave abnormality
   2 = left ventricular hyperthrophy

8. ***Max heart rate achieved*** : displays the max heart rate achieved by an individual.

9. ***Exercise induced angina*** :

   1 = yes

   0 = no

10. ***ST depression induced by exercise relative to rest***: displays the value which is an integer or float.

11. ***Peak exercise ST segment*** :

   1 = upsloping

   2 = flat

   3 = downsloping

12. ***Number of major vessels (0–3) colored by flourosopy*** : displays the value as integer or float.

13. ***Thal*** : displays the thalassemia :

    3 = normal

    6 = fixed defect

    7 = reversible defect

14. ***Diagnosis of heart disease*** : Displays whether the individual is suffering from heart disease or not :

    0 = absence

    1, 2, 3, 4 = present.

1. **Age**: Age is the most important risk factor in developing cardiovascular or heart diseases, with approximately a tripling of risk with each decade of life. Coronary fatty streaks can begin to form in adolescence. It is estimated that 82 percent of people who die of coronary heart disease are 65 and older. Simultaneously, the risk of stroke doubles every decade after age 55.

2. **Sex**: Men are at greater risk of heart disease than pre-menopausal women. Once past menopause, it has been argued that a woman's risk is similar to a man's although more recent data from the WHO and UN disputes this. If a female has diabetes, she is more likely to develop heart disease than a male with diabetes.

3. **Angina (Chest Pain)**: Angina is chest pain or discomfort caused when your heart muscle doesn't get enough oxygen-rich blood. It may feel like pressure or squeezing in your chest. The discomfort also can occur in your shoulders, arms, neck, jaw, or back. Angina pain may even feel like indigestion.

4. **Resting Blood Pressure:** Over time, high blood pressure can damage arteries that feed your heart. High blood pressure that occurs with other conditions, such as obesity, high cholesterol or diabetes, increases your risk even more.

5. **Serum Cholesterol**: A high level of low-density lipoprotein (LDL) cholesterol (the “bad” cholesterol) is most likely to narrow arteries. A high level of triglycerides, a type of blood fat related to your diet, also ups your risk of a heart attack. However, a high level of high-density lipoprotein (HDL) cholesterol (the “good” cholesterol) lowers your risk of a heart attack.

6. **Fasting Blood Sugar**: Not producing enough of a hormone secreted by your pancreas (insulin) or not responding to insulin properly causes your body's blood sugar levels to rise, increasing your risk of a heart attack.

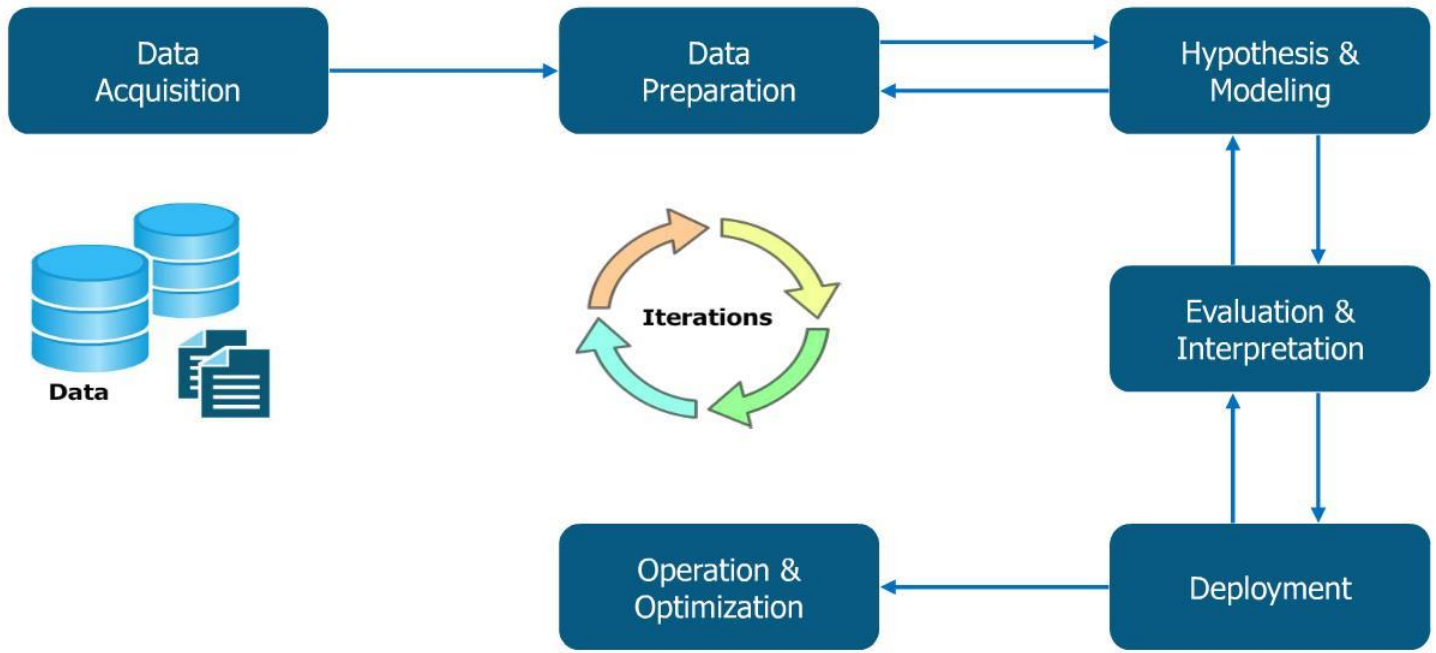
7. **Resting ECG**: For people at low risk of cardiovascular disease, the USPSTF concludes with moderate certainty that the potential harms of screening with resting or exercise ECG equal or exceed the potential benefits. For people at intermediate to high risk, current evidence is insufficient to assess the balance of benefits and harms of screening.

8. **Max heart rate achieved**: The increase in cardiovascular risk, associated with the acceleration of heart rate, was comparable to the increase in risk observed with high blood pressure. It has been shown that an increase in heart rate by 10 beats per minute was associated with an increase in the risk of cardiac death by at least 20%, and this increase in the risk is similar to the one observed with an increase in systolic blood pressure by 10 mm Hg.

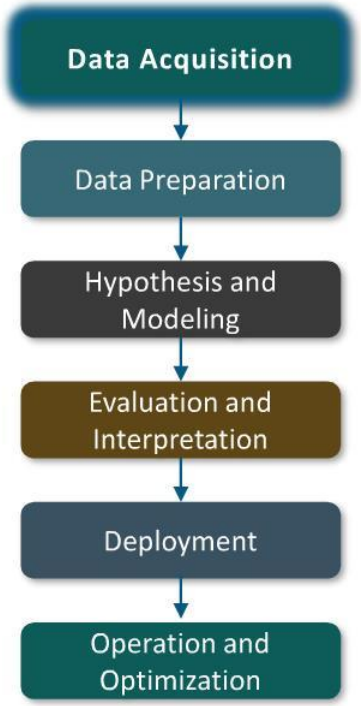
9. **Exercise induced angina**: The pain or discomfort associated with angina usually feels tight, gripping or squeezing, and can vary from mild to severe. Angina is usually felt in the center of your chest but may spread to either or both of your shoulders, or your back, neck, jaw or arm. It can even be felt in your hands. o Types of Angina a. Stable Angina / Angina Pectoris b. Unstable Angina c. Variant (Prinzmetal) Angina d. Microvascular Angina.

10. **Peak exercise ST segment**: A treadmill ECG stress test is considered abnormal when there is a horizontal or down-sloping ST-segment depression ≥ 1 mm at 60–80 ms after the J point. Exercise ECGs with up-sloping ST-segment depressions are typically reported as an 'equivocal' test. In general, the occurrence of horizontal or down-sloping ST-segment depression at a lower workload (calculated in METs) or heart rate indicates a worse prognosis and higher likelihood of multi-vessel disease. The duration of ST-segment depression is also important, as prolonged recovery after peak stress is consistent with a positive treadmill ECG stress test. Another finding that is highly indicative of significant CAD is the occurrence of ST-segment elevation > 1 mm (often suggesting transmural ischemia); these patients are frequently referred urgently for coronary angiography.

Life Cycle of Data Science



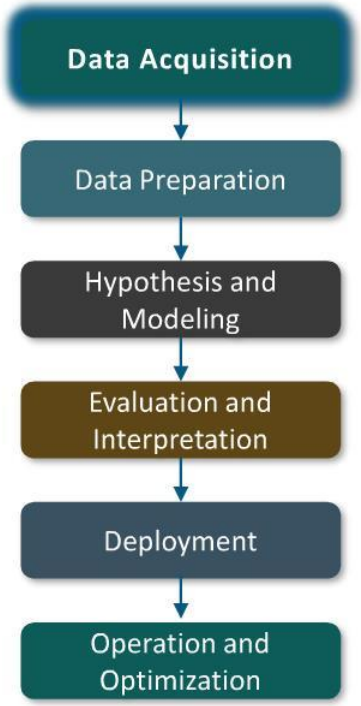
Data Acquisition/ Data Collection



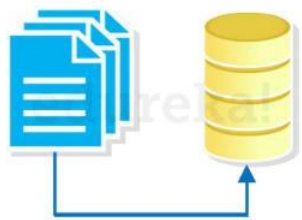
Now to help John let's see what data we can collect from different locations and how it affects the pricing of an apartment

Price	Apartment name/no.	No of bedrooms	Floor number	Criminal rate per year	Pollution level	Distance to nearby Educational institution
30L	xv	3	2	3	15	900 m
20L	cs	2	4	2		2 km
28L	df	2	G	5	13	1.5 km
25L	re	1	3	1	12	1.7 m
30L	sd	2	0	3	13	700 m

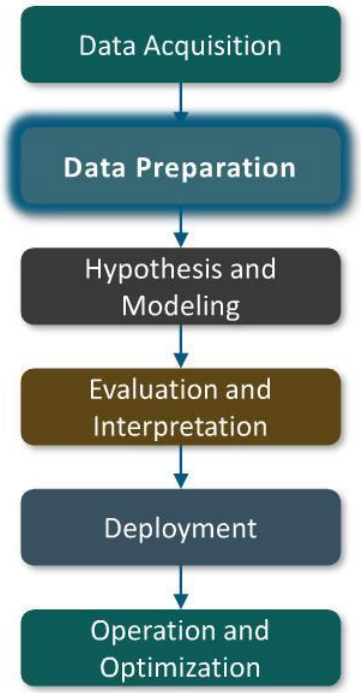
Data Acquisition/ Data Collection



- Data acquisition involves acquiring data from all the identified internal and external sources that can help answer the business question
- This data could be
  - logs from webservers
  - social media data
  - census datasets
  - data streamed from online sources via APIs



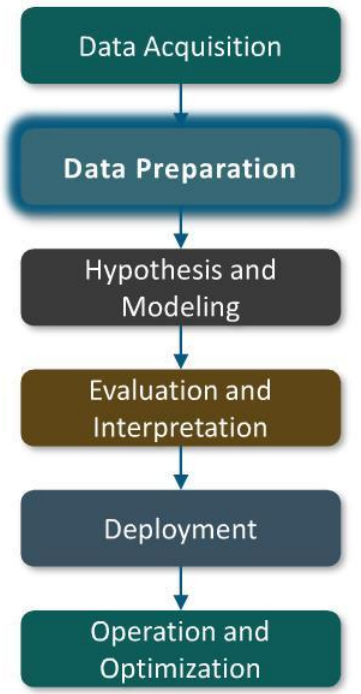
Data Preparation



- The data we have collected is not clean, there are some errors which need to cleansed
- Also we may need to change the values of columns as per requirements

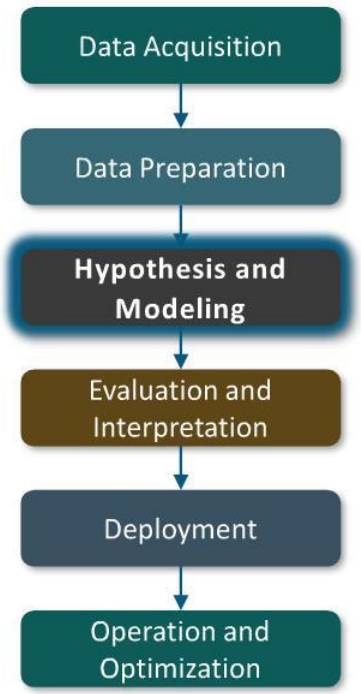
Price	Apartment name/no.	No of bedrooms	Floor number	Criminal rate per year	Pollution level	Educational institution within 1km radius
30L	xv	3	2	3	15	Yes
20L	cs	2	4	2		No
28L	df	2	G	5	13	No
25L	re	1	3	1	12	No
30L	sd	2	0	3	13	Yes

Data Preparation

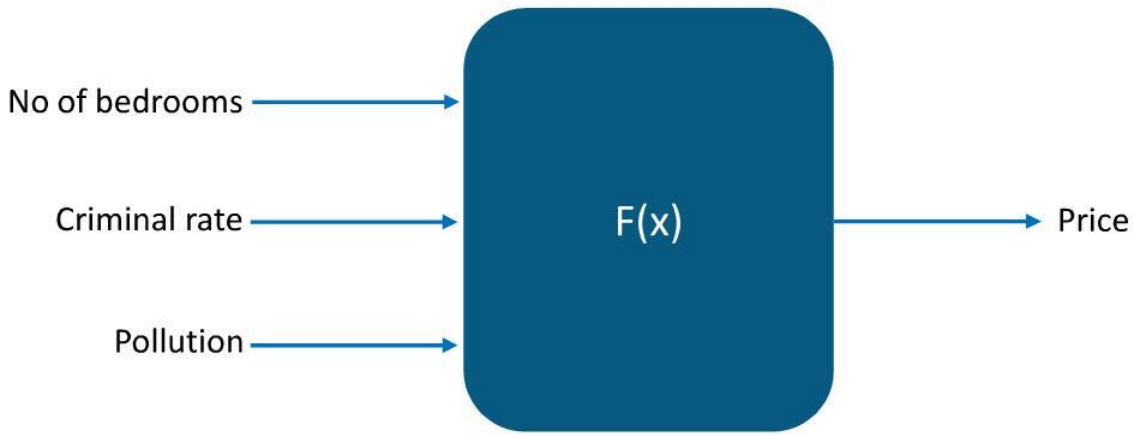


- Data Wrangling is the process of cleaning and unifying messy and complex data sets
- Data after reformatting can be converted to JSON, CSV or any other format that makes it easy to load into one of the data science tools

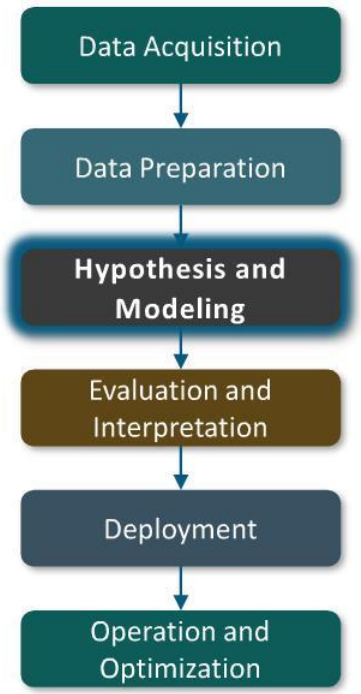
Hypothesis and Modeling



Based on the requirements, a model is created using the dataset

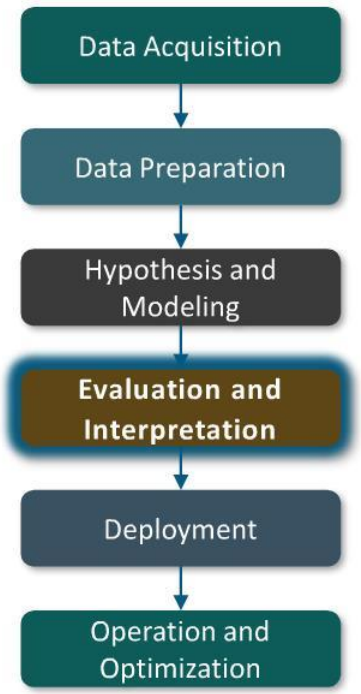


Hypothesis and Modeling

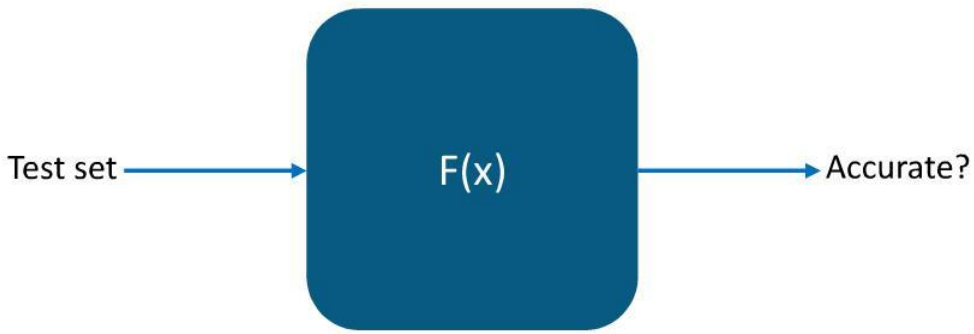


- Involves forming and testing hypotheses about the data and the processes that generate it
- Requires writing, running and refining the programs to analyze and derive meaningful business insights from data
- Mostly written in languages like Python, R, Spark

Evaluation and Interpretation

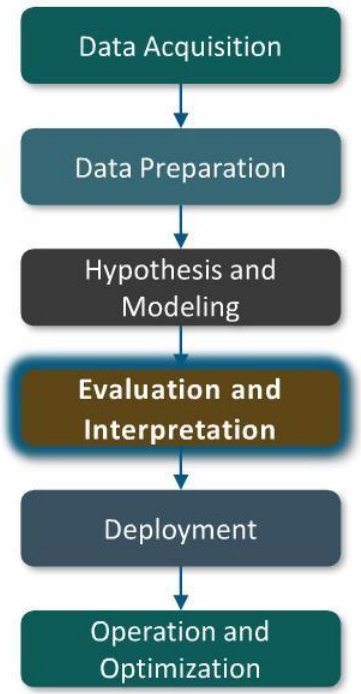


This model is evaluated using test data set



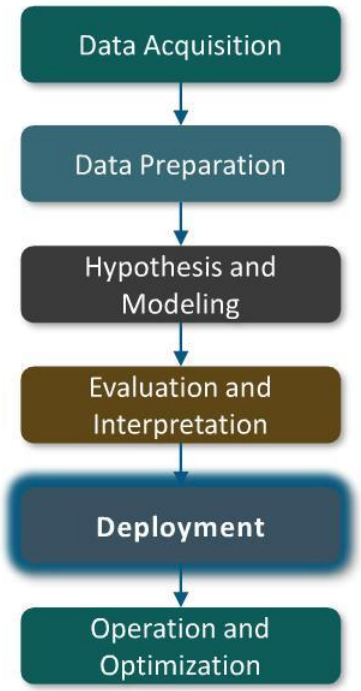
If accuracy is low, the above steps are repeated until a good model is found

Evaluation and Interpretation

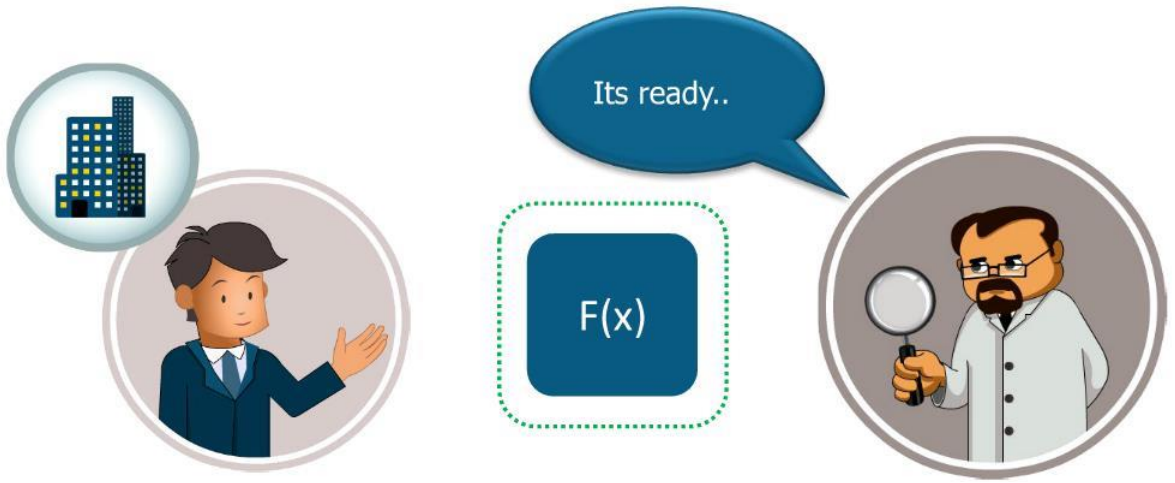


- Model performances should be measured and compared using validation and test datasets
- Models should have a high accuracy for implementation

Deployment



- In this step the model we created is deployed in to the market
- Models generally have to be recoded before deployment (e.g., data scientists may favor Python, but production environments may require Java)



- Exploratory Data Analysis (EDA) is a pre-processing step to understand the data. There are numerous methods and steps in performing EDA,

Cardiovascular diseases (CVDs) or heart disease are the number one cause of death globally with 17.9 million death cases each year. CVDs are concertedly contributed by hypertension, diabetes, overweight and unhealthy lifestyles.

The outline for EDA are as follows;

1. Import and get to know the data

2. Data Cleaning

> *a) Check the data type*
>
> *b) Check for the data characters mistakes*
>
> *c) Check for missing values and replace them*
>
> *d) Check for duplicate rows*
>
> *e) Statistics summary*
>
> *f) Outliers and how to remove them*

3. Distributions and Relationship

a) Categorical variable distribution

b) Continuous variable distribution

c) Relationship between categorical and continuous variables

Variables or features explanations:

1. age (Age in years)
2. sex : (1 = male, 0 = female)
3. cp (Chest Pain Type): [0: Typical Angina, 1: Atypical Angina, 2: Non-Anginal Pain, 3: Asymptomatic]
4. trestbps (Resting Blood Pressure in mm/hg)

5. chol (Serum Cholesterol in mg/dl)
6. fps (Fasting Blood Sugar > 120 mg/dl): [0 = no, 1 = yes]
7. restecg (Resting ECG): [0: normal, 1: having ST-T wave abnormality , 2: showing probable or definite left ventricular hypertrophy]
8. thalach (maximum heart rate achieved)
9. exang (Exercise Induced Angina): [1 = yes, 0 = no]
10. oldpeak (ST depression induced by exercise relative to rest)

11. slope (the slope of the peak exercise ST segment)

12. ca [number of major vessels (0–3)]

13. thal : [1 = normal, 2 = fixed defect, 3 = reversible defect]

14. target: [0 = disease, 1 = no disease]

2. Data Cleaning

> *a) Check the data type.*

The variables types are

- Binary: sex, fbs, exang, target
- Categorical: cp, restecg, slope, ca, thal
- Continuous: age, trestbps, chol, thalac, oldpeak

b. Check for the data characters mistakes

1. feature **'ca'** ranges from 0–3, however, `df.nunique()` listed 0–4. So lets find the '4' and change them to NaN.

```
[ ] df['ca'].unique()
```

```
array([0, 2, 1, 3, 4], dtype=object)
```

```
# to count the number in of each category decending order
df.ca.value_counts()
```

```
0    175
1     65
2     38
3     20
4      5
Name: ca, dtype: int64
```

2. Feature **'thal'** ranges from 1–3, however, `df.nunique()` listed 0–3. There are two values of '0'. So lets change them to NaN.

```
[5] df.thal.value_counts()
```

```
2    166
3    117
1     18
0      2
Name: thal, dtype: int64
```

```
[10] df.loc[df['thal']==0, 'thal'] = np.NaN
```

```
[11] df[df['thal']==0]
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target

```
[12] df['thal'].unique()
```

c) Check for missing values and replace them

```
# to check missing values
df.isnull().sum()
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          5
thal        2
target      0
dtype: int64
```

d) Check for duplicate rows

```
[ ] duplicated = df.duplicated().sum()
    if duplicated:
        print('Duplicates Rows in Dataset are : {}'.format(duplicated))
    else:
        print('Dataset contains no Duplicate Values')
```

```
Duplicates Rows in Dataset are : 1
```

```
[ ] duplicated = df[df.duplicated(keep=False)]
    duplicated.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
163	38	1	2	138	175	0	1	173	0	0.0	2	0.0	2.0	1
164	38	1	2	138	175	0	1	173	0	0.0	2	0.0	2.0	1

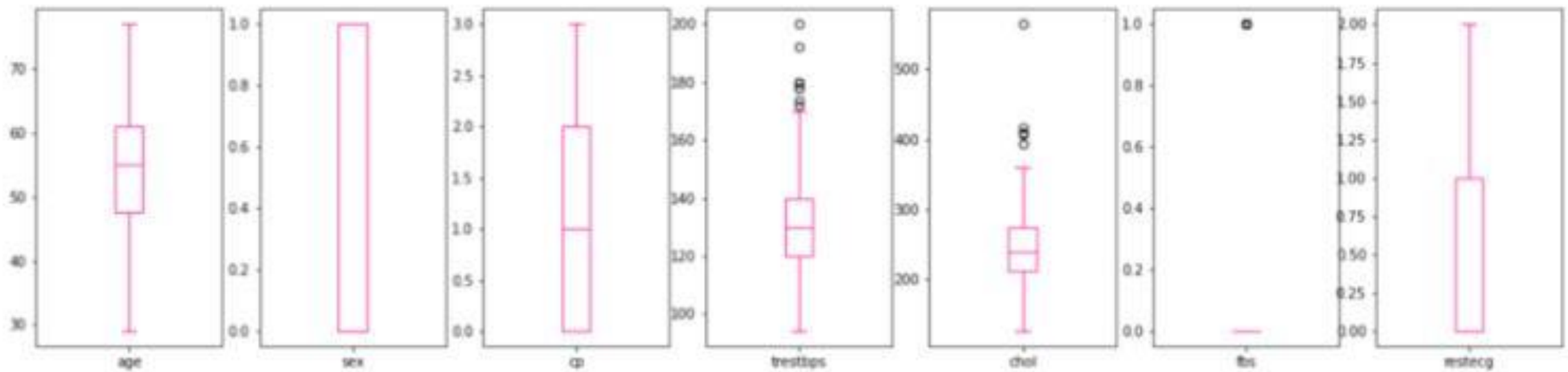
Image snapshot by author

Basically, with `df.describe()`, we should check on the min and max value for the categorical variables (min-max). Sex (0–1), cp (0–3), fbs (0–1), restecg (0–2), exang (0–1), slope (0–2), ca (0–3), thal (0–3). We should also observe the mean, std, 25% and 75% on the continuous variables.

f) Outliers and how to remove them

```
1   df.plot(kind='box', subplots=True, layout=(2,7),
2   sharex=False,sharey=False, figsize=(20, 10),
3   color='deeppink');
```

Outliers plotting for Exploratory Data Analysis hosted with ❤ by GitHub view raw



Machine Learning

- Teach machines to solve problems
- Evaluated using performance measures
- What is 95% accuracy?
    - Classification: 95 / 100 shoes correctly classified
    - Regression:Predict 95/100 house prices correctly

$600,000

$400,000 ✗

$599,999 ✗

?

Performance Measures

Classification: Predict Category

- Simple Accuracy
- Precision
- Recall
- F-beta measure
- ROC (and AUC)

Regression: Predict Value

- Sum of Squares Error
- RMS Error
- Mean Absolute Error

Confusion Matrix

Nike	Not Nike

Confusion Matrix

Nike

Not Nike

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

Confusion Matrix

- True Positive: Predict Nike shoe as Nike
- False Positive: Predict Non-Nike shoe as Nike
- False Negative: Predict Nike shoe as Non-Nike
- True Negative: Predict Non-Nike shoe as Non-Nike

	p' (Predicted)	n' (Predicted)
P (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

Simple Accuracy

$$Accuracy = \frac{No.Samples\ Predicted\ Correctly}{Total\ No.of\ Samples}$$

What is wrong with this ?

9,990 Non-Nike

10 Nike

```
def classifier(shoe):
    return False
```

$$Accuracy = \frac{9{,}990}{10{,}000} = \mathbf{99.9\%}$$

Performance Measures

Precision: Of the shoes classified Nike, How many are actually Nike ?

Recall: Of the shoes that are actually Nike, How many are classified as Nike ?

Performance Measures

Precision: Of the shoes classified Nike, How many are actually Nike ?

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

1. Number of shoes classified Nike

   = TP + FP

2. Number of shoes actually Nike (when classified as nike)

   = TP

$$Precision = \frac{TP}{TP + FP}$$

Performance Measures

Recall: Of the shoes that are actually Nike, How many are classified as Nike ?

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

1. Number of shoes actually Nike

   = TP + FN

2. Number of shoes classified Nike (when actually Nike)

   = TP

$$Recall = \frac{TP}{TP + FN}$$

Performance Measures

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{0}{0}\ (Not\ Defined)$$

$$Recall = \frac{0}{0 + 10} = 0$$

$$Simple\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 99.9\%$$

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive 0	False Negative 10
n (Actual)	False Positive 0	True Negative 9,990

Performance Measures

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{0}{0}\ (Not\ Defined)$$

$$Recall = \frac{0}{0 + 10} = 0$$

$$Simple\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 99.9\%$$

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive 0	False Negative 10
n (Actual)	False Positive 0	True Negative 9,990

Performance Measures

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

```
def classifier(shoe):
    return False True
```

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive 10	False Negative 0
n (Actual)	False Positive 9,990	True Negative 0

Performance Measures

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

```
def classifier(shoe):
    return ~~False~~ True
```

$$Simple\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.1\%$$

$$Precision = \frac{10}{10 + 9{,}990} = 0.001$$

$$Recall = \frac{10}{10 + 0} = 1.0$$

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive 10	False Negative 0
n (Actual)	False Positive 9,990	True Negative 0

Performance Measures

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

```
def classifier(shoe):
    return ~~False~~ True
```

$$Simple\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.1\%$$

$$Precision = \frac{10}{10 + 9{,}990} = 0.001$$

$$Recall = \frac{10}{10 + 0} = 1.0$$

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive 10	False Negative 0
n (Actual)	False Positive 9,990	True Negative 0

Comparing Systems

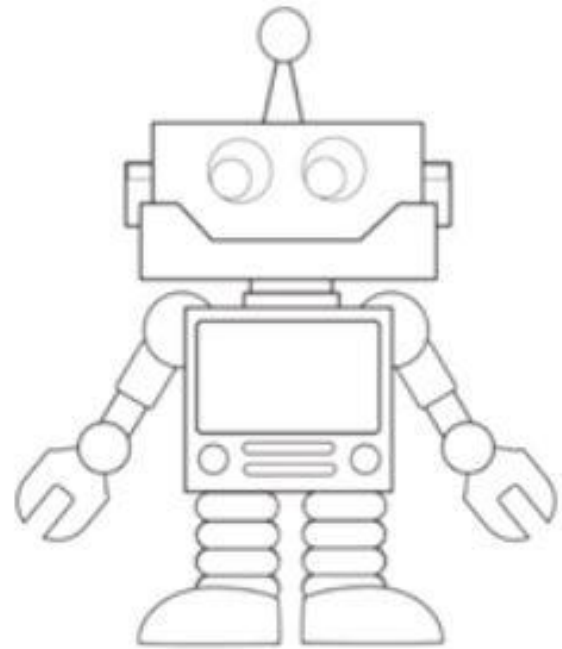
System 1
- Precision: 70%
- Recall: 60%

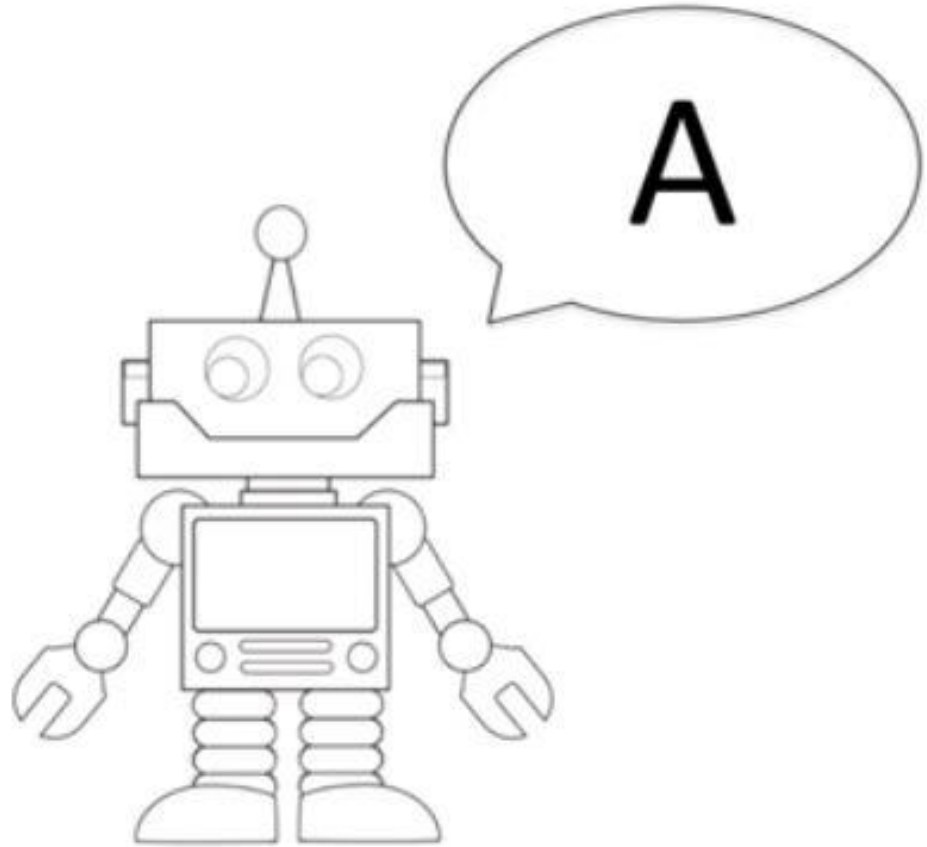
System 2
- Precision: 80%
- Recall: 50%

$$F_\beta = \frac{1}{\beta \times \frac{1}{Precision} + (1-\beta) \times \frac{1}{Recall}}$$

- Greater β, Greater importance to Precision
- Cancer classification requires Precision, hence high β
- Making sure Nike shoes are classified as such requires Higher Recall, hence lower β

A



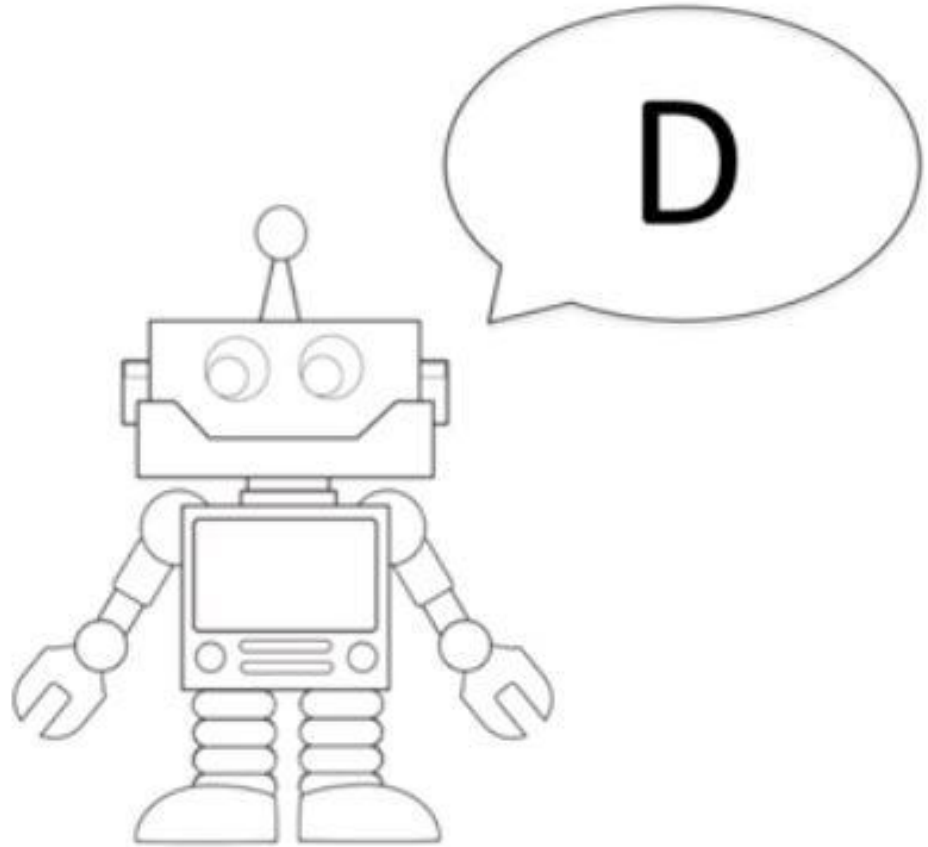


A

answers →

input ↓

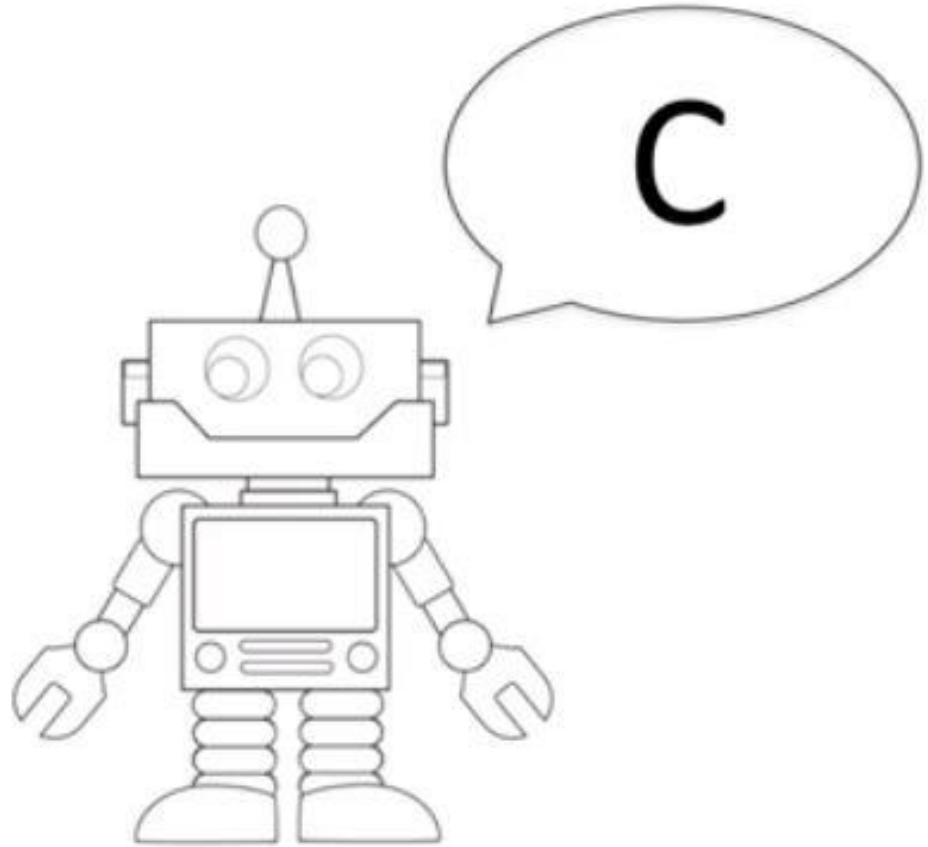
	A	B	C	D
A	1			
B				
C				
D				



answers →

input ↓

	A	B	C	D
A	1			
B				1
C				
D				

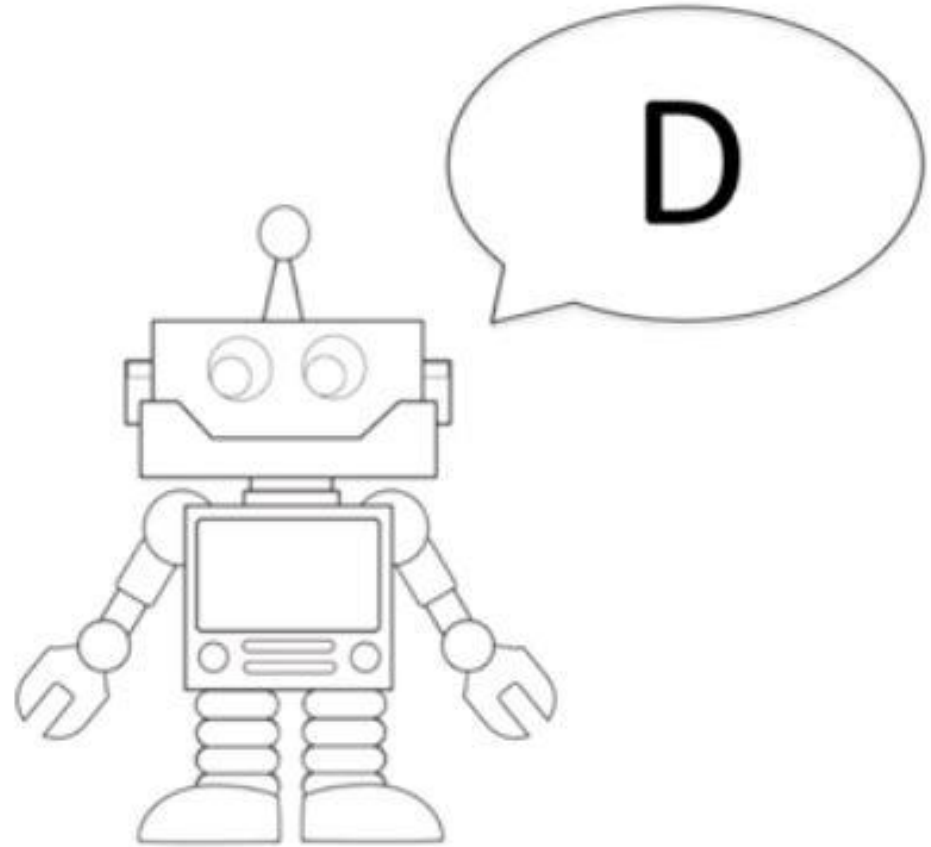


C

answers →

input ↓

	A	B	C	D
A	1			
B				1
C			1	
D				

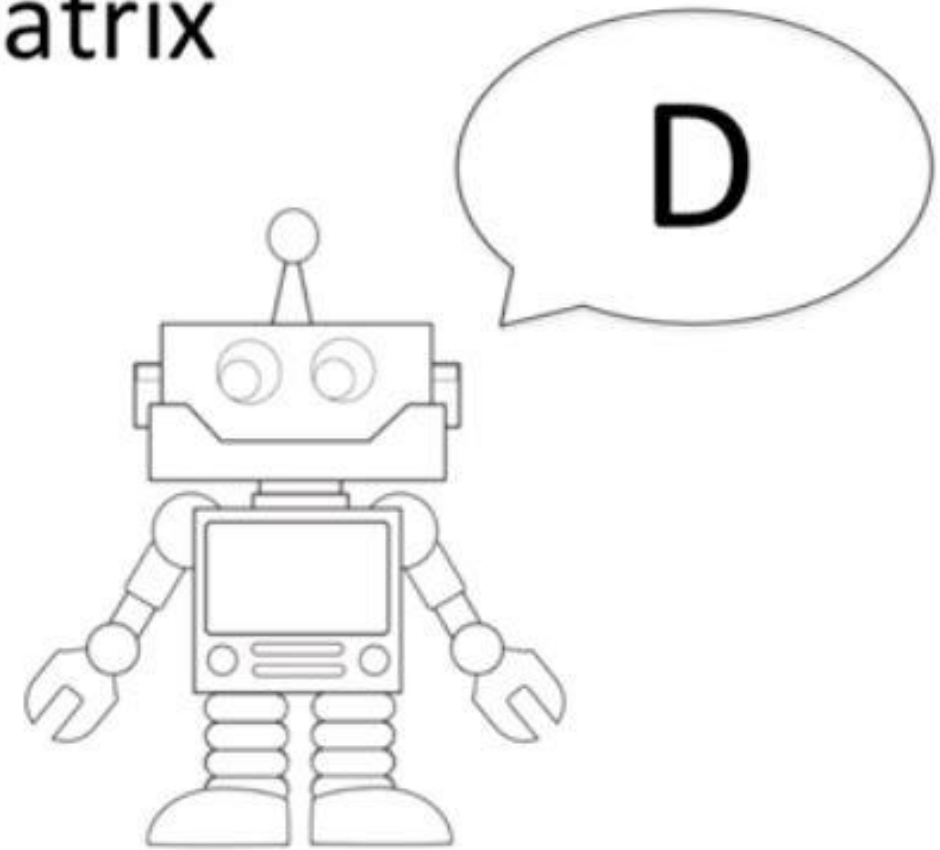


answers →

input ↓

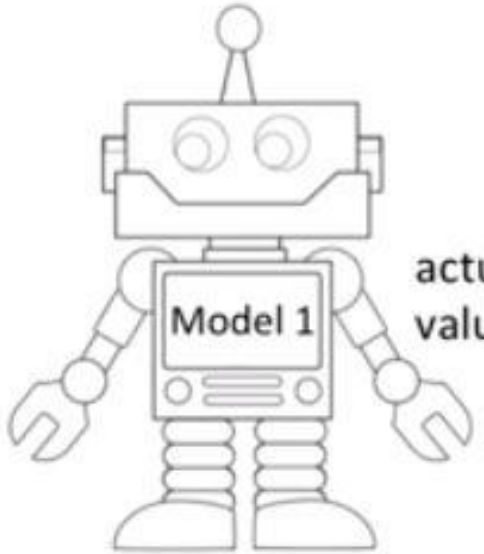
	A	B	C	D
A	1			
B				1
C			1	
D				2

Confusion Matrix



	A	B	C	D
A	1	0	0	0
B	0	1	0	1
C	0	0	1	0
D	0	0	0	2

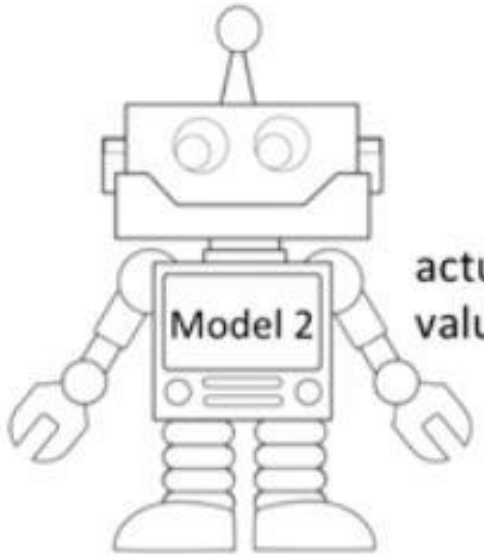
Which model performs better?



actual values

predictions →

	A	B	C	D
A	10	0	0	0
B	0	5	3	2
C	0	1	8	1
D	0	1	0	9



actual values

predictions →

	A	B	C	D
A	8	2	0	0
B	1	7	0	2
C	0	0	9	1
D	2	3	0	5

Performance measures

- Accuracy
- Precision
- Recall
- F1 score

Performance measures

- TP (true positive)
- TN (true negative)
- FP (false positive)
- FN (false negative)

predictions →

actual values ↓

	A	B	C	D
A	9	1	0	0
B	1	15	3	1
C	5	0	24	1
D	0	4	1	15

True Positive

predictions (output) →

actual class (input) ↓

	A	B	C	D
A	9	1	0	0
B	1	15	3	1
C	5	0	24	1
D	0	4	1	15

correctly identified prediction for each class

True Negative for A

predictions (output) →

actual class (input) ↓

	A	B	C	D
A	9	1	0	0
B	1	15	3	1
C	5	0	24	1
D	0	4	1	15

correctly rejected prediction for certain class (A)

True Negative for D

predictions (output) →

actual class (input) ↓

	A	B	C	D
A	9	1	0	0
B	1	15	3	1
C	5	0	24	1
D	0	4	1	15

correctly rejected prediction for certain class (D)

False Positive for A

predictions (output) →

actual class (input) ↓

	A	B	C	D
A	9	1	0	0
B	1	15	3	1
C	5	0	24	1
D	0	4	1	15

incorrectly identified predictions for certain class (A)

False Positive for B

predictions (output) →

actual class (input) ↓

	A	B	C	D
A	9	1	0	0
B	1	15	3	1
C	5	0	24	1
D	0	4	1	15

incorrectly identified predictions for certain class (B)

False Negative for A

predictions (output) →

actual class (input) ↓

	A	B	C	D
A	9	1	0	0
B	1	15	3	1
C	5	0	24	1
D	0	4	1	15

incorrectly rejected for certain class (A)

Accuracy

- Accuracy is calculated as the total number of correct predictions divided by the total number of dataset

Accuracy

predictions (output) →

actual class (input) ↓

	A	B	C	D
A	9	1	0	0
B	1	15	3	1
C	5	0	24	1
D	0	4	1	15

correctly identified prediction for each class
9 + 15 + 24 + 15

Accuracy

predictions (output) →

actual class (input) ↓

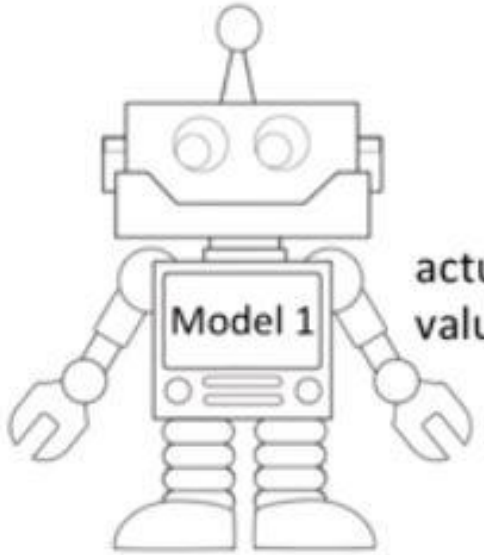
	A	B	C	D
A	9	1	0	0
B	1	15	3	1
C	5	0	24	1
D	0	4	1	15

correctly identified prediction for each class / total dataset

9 + 15 + 24 + 15 / 80

accuracy = 0.78

Accuracy Comparison

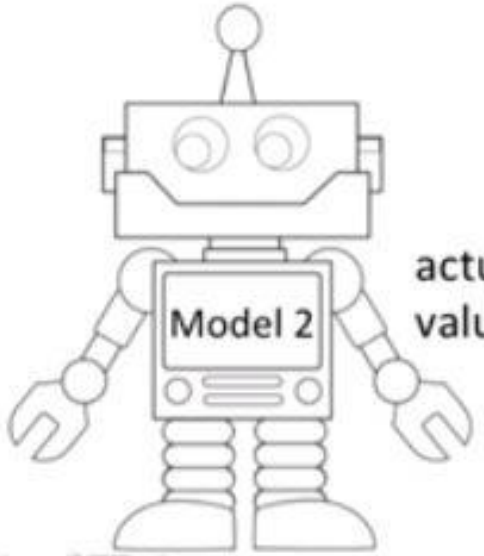


actual values

predictions →

	A	B	C	D
A	10	0	0	0
B	0	5	3	2
C	0	1	8	1
D	0	1	0	9

(10 + 5 + 8 + 9) / 40 = **0.8**



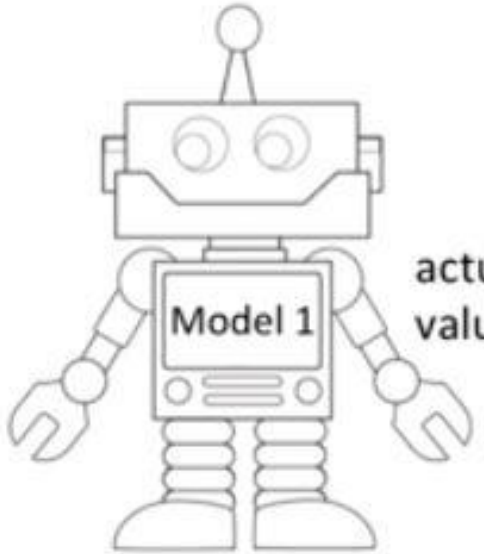
actual values

predictions →

	A	B	C	D
A	8	2	0	0
B	1	7	0	2
C	0	0	9	1
D	2	3	0	5

(8 + 7 + 9 + 5) / 40 = **0.725**

Accuracy works well on balanced data

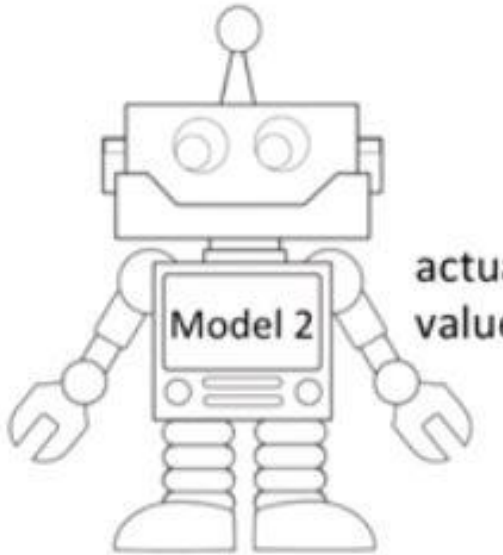


actual values

predictions →

	A	B	C	D
A	10	0	0	0
B	0	5	3	2
C	0	1	8	1
D	0	1	0	9

(10 + 5 + 8 + 9) / 40 = **0.8**



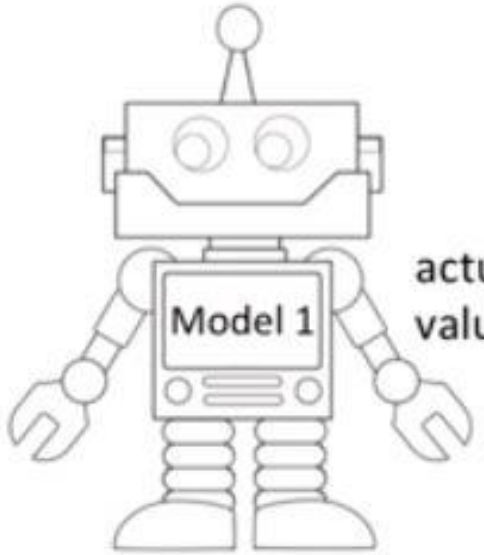
actual values

predictions →

	A	B	C	D
A	8	2	0	0
B	1	7	0	2
C	0	0	9	1
D	2	3	0	5

(8 + 7 + 9 + 5) / 40 = **0.725**

Accuracy on imbalanced data misleads performance

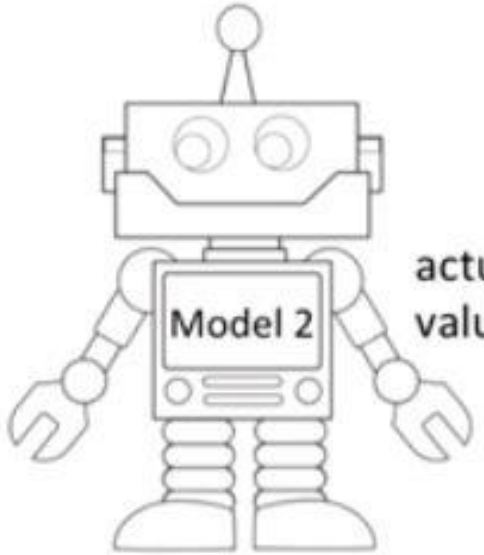


actual values ↓

predictions →

	A	B	C	D
A	100	80	10	10
B	0	9	0	1
C	0	1	8	1
D	0	1	0	9

(100 + 9 + 8 + 9) / 230 = **0.547**



actual values ↓

predictions →

	A	B	C	D
A	198	2	0	0
B	7	1	0	2
C	0	8	1	1
D	2	3	4	1

(198 + 1 + 1 + 1) / 230 = **0.87**

F1 score is good metric when data is imbalanced

Given a class, will the classifier detect it ? (recall)

	A	B	C	D
A	100	80	10	10
B	0	9	0	1
C	0	1	8	1
D	0	1	0	9

Given a class prediction from the classifier,
how likely is it to be correct? (precision)

F1 score is good metric when data is imbalanced

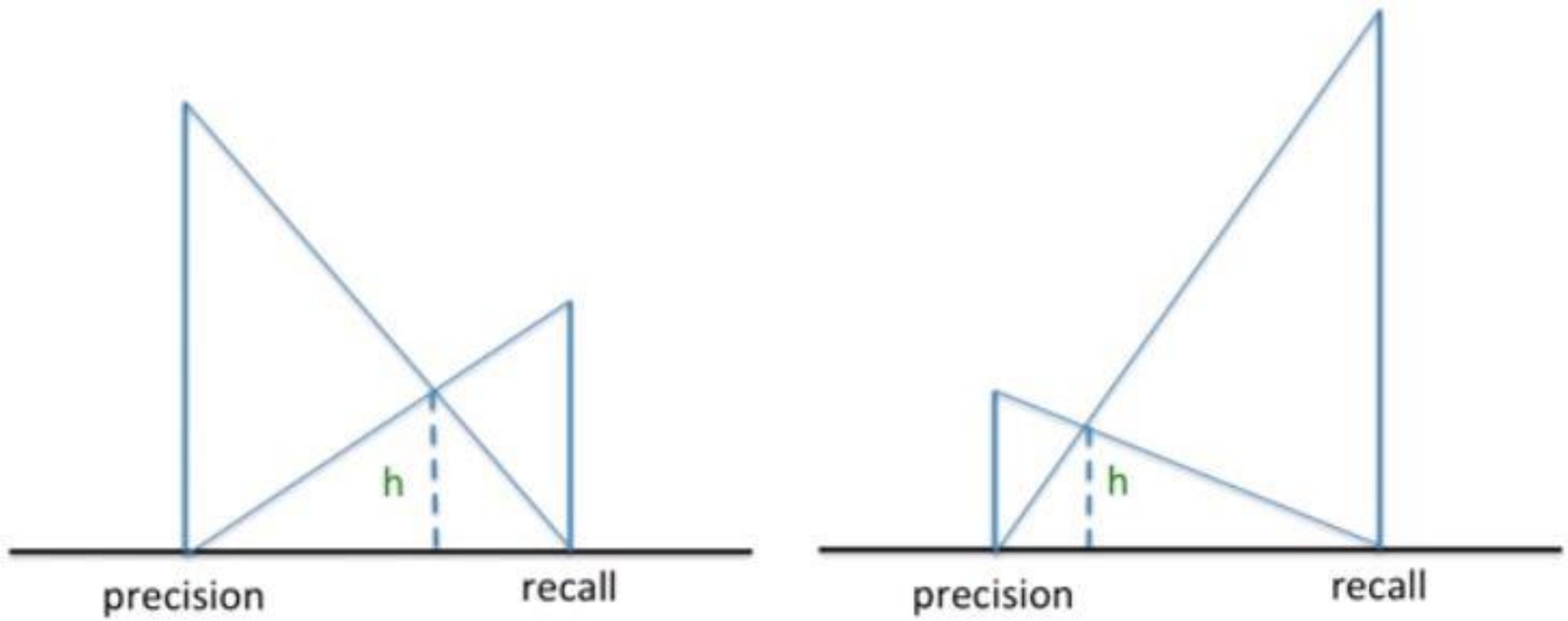
Given a class, will the classifier detect it ? (recall)

	A	B	C	D
A	100	80	10	10
B	0	9	0	1
C	0	1	8	1
D	0	1	0	9

Given a class prediction from the classifier,
how likely is it to be correct? (precision)

F1 Score is harmonic mean of recall and precision

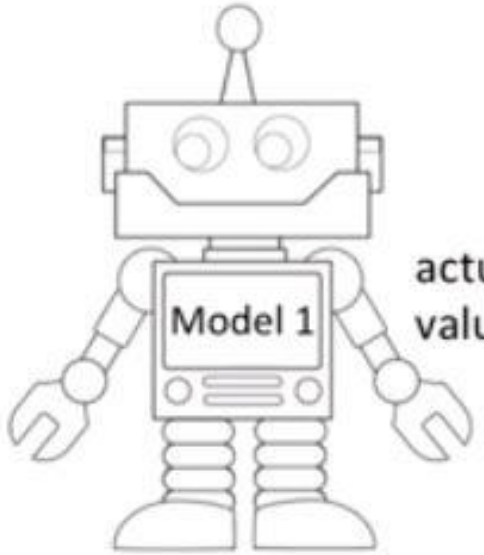
Harmonic Mean punishes extreme value more



h is half the harmonic mean

$$\text{F1 Score} = 2 \times \frac{\text{Precision X Recall}}{\text{Precision + Recall}}$$

Precision of Model 1 (macro average)



actual values

predictions →

	A	B	C	D
A	100	80	10	10
B	0	9	0	1
C	0	1	8	1
D	0	1	0	9
	TP: 100 FP: 0	TP: 9 FP: 82	TP: 8 FP: 10	TP: 9 FP: 12

Precision = TP / (TP + FP) P(A) = 1 P(B) = 9/91 P(C) = 8/18 P(D) = 9/21

average precision = P(A) + P(B) + P(C) + P(D) / 4 = 0.492

the number of classes

Recall of Model 1 (macro average)

predictions →

	A	B	C	D
A	100	80	10	10
B	0	9	0	1
C	0	1	8	1
D	0	1	0	9

TP: 100, FN: 100 R(A) = 100 / 200

TP: 9, FN: 1 R(B) = 9/10

TP: 8, FN: 2 R(C) = 8/10

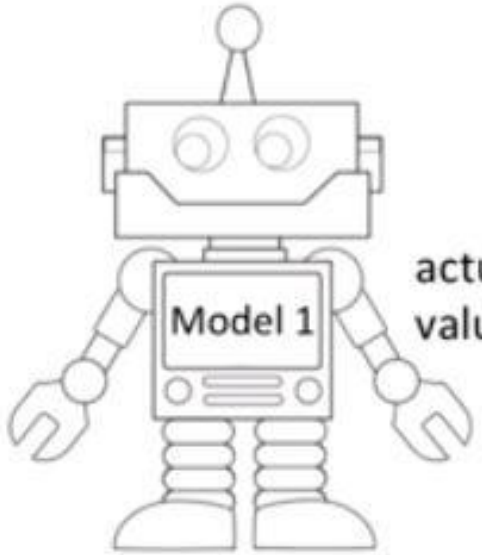
TP: 9, FN: 1 R(D) = 9/10

Recall = TP / (TP + FN)

average recall = R(A) + R(B) + R(C) + R(D) / 4 = 0.775

the number of classes

F1 Score of Model 1



actual values ↓

predictions →

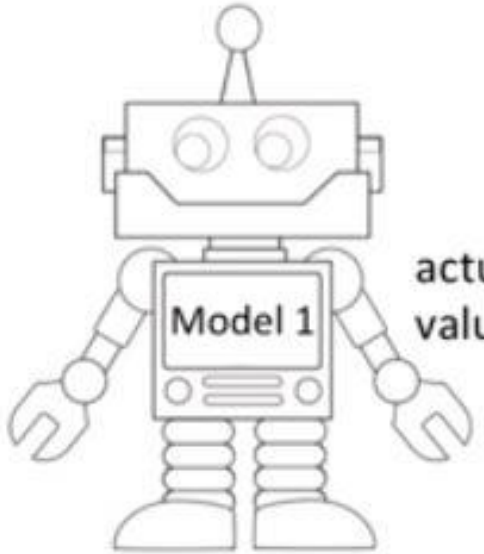
	A	B	C	D
A	100	80	10	10
B	0	9	0	1
C	0	1	8	1
D	0	1	0	9

$$\text{F1 Score} = 2 \text{ X } \frac{\text{Precision X Recall}}{\text{Precision + Recall}}$$

$$2 \text{ X } \frac{0.492 \text{ X } 0.775}{0.492 + 0.775}$$

0.601

F1 Score on imbalanced data



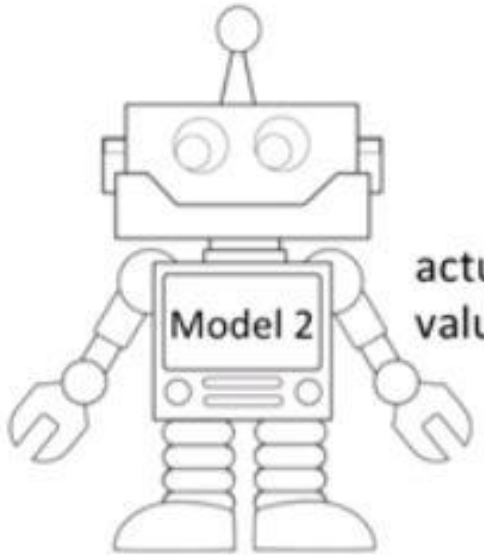
actual values

predictions →

	A	B	C	D
A	100	80	10	10
B	0	9	0	1
C	0	1	8	1
D	0	1	0	9

F1 Score = **0.601**

accuracy = 0.547



actual values

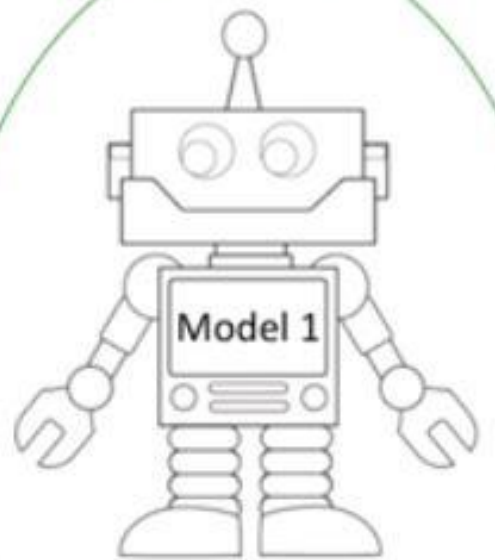
predictions →

	A	B	C	D
A	198	2	0	0
B	7	1	0	2
C	0	8	1	1
D	2	3	4	1

F1 Score = **0.342**

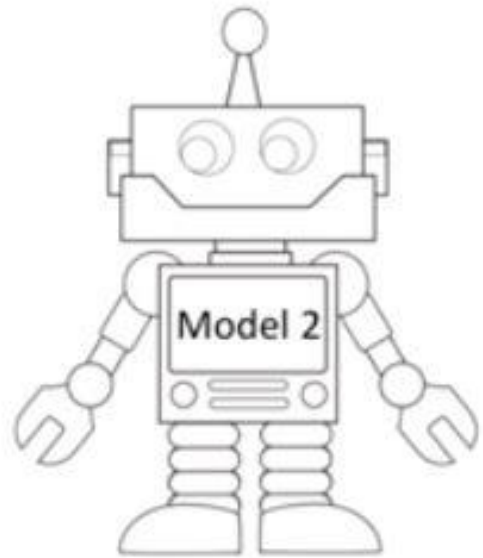
accuracy = 0.87

F1 Score on imbalanced data



F1 Score = **0.601**

accuracy = 0.547



F1 Score = **0.342**

accuracy = 0.87

Model 1 predicts well on multiple class classification on imbalanced given data, and F1 score is the metric to quantify its performance.

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

- There are two possible predicted classes: "yes" and "no". If we were predicting the presence of a disease, for example, "yes" would mean they have the disease, and "no" would mean they don't have the disease.
- The classifier made a total of 165 predictions (e.g., 165 patients were being tested for the presence of that disease).
- Out of those 165 cases, the classifier predicted "yes" 110 times, and "no" 55 times.

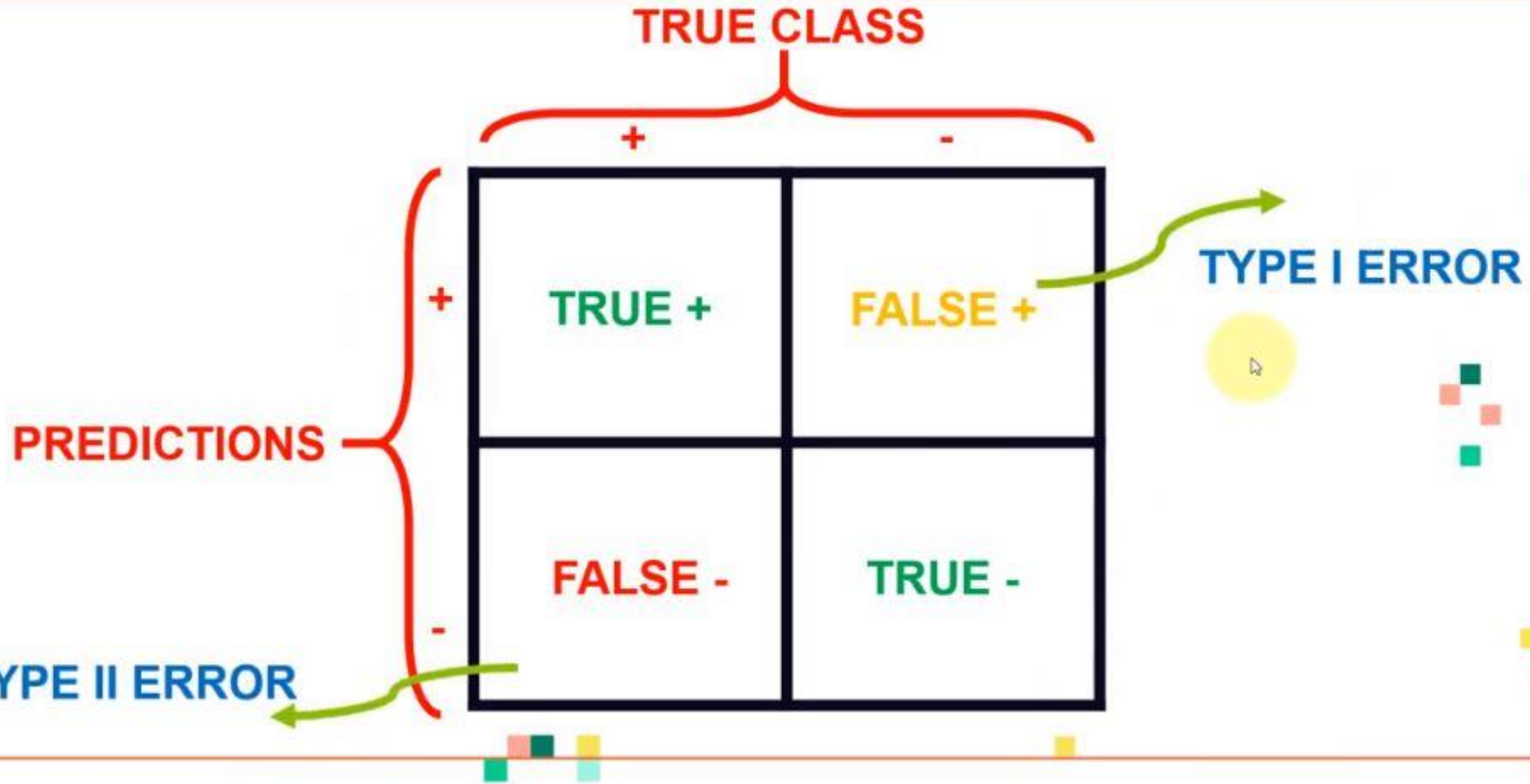
This is a list of rates that are often computed from a confusion matrix for a binary classifier:

- **Accuracy:** Overall, how often is the classifier correct?
    - (TP+TN)/total = (100+50)/165 = 0.91
- **Misclassification Rate:** Overall, how often is it wrong?
    - (FP+FN)/total = (10+5)/165 = 0.09
    - equivalent to 1 minus Accuracy
    - also known as "Error Rate"
- **True Positive Rate:** When it's actually yes, how often does it predict yes?
    - TP/actual yes = 100/105 = 0.95
    - also known as "Sensitivity" or "Recall"
- **False Positive Rate:** When it's actually no, how often does it predict yes?
    - FP/actual no = 10/60 = 0.17
- **True Negative Rate:** When it's actually no, how often does it predict no?
    - TN/actual no = 50/60 = 0.83
    - equivalent to 1 minus False Positive Rate
    - also known as "Specificity"

- (TP+TN)/total = (100+50)/165 = 0.91

- **Misclassification Rate:** Overall, how often is it wrong?
  - (FP+FN)/total = (10+5)/165 = 0.09
  - equivalent to 1 minus Accuracy
  - also known as "Error Rate"
- **True Positive Rate:** When it's actually yes, how often does it predict yes?
  - TP/actual yes = 100/105 = 0.95
  - also known as "Sensitivity" or "Recall"
- **False Positive Rate:** When it's actually no, how often does it predict yes?
  - FP/actual no = 10/60 = 0.17
- **True Negative Rate:** When it's actually no, how often does it predict no?
  - TN/actual no = 50/60 = 0.83
  - equivalent to 1 minus False Positive Rate
  - also known as "Specificity"
- **Precision:** When it predicts yes, how often is it correct?
  - TP/predicted yes = 100/110 = 0.91

CONFUSION MATRIX



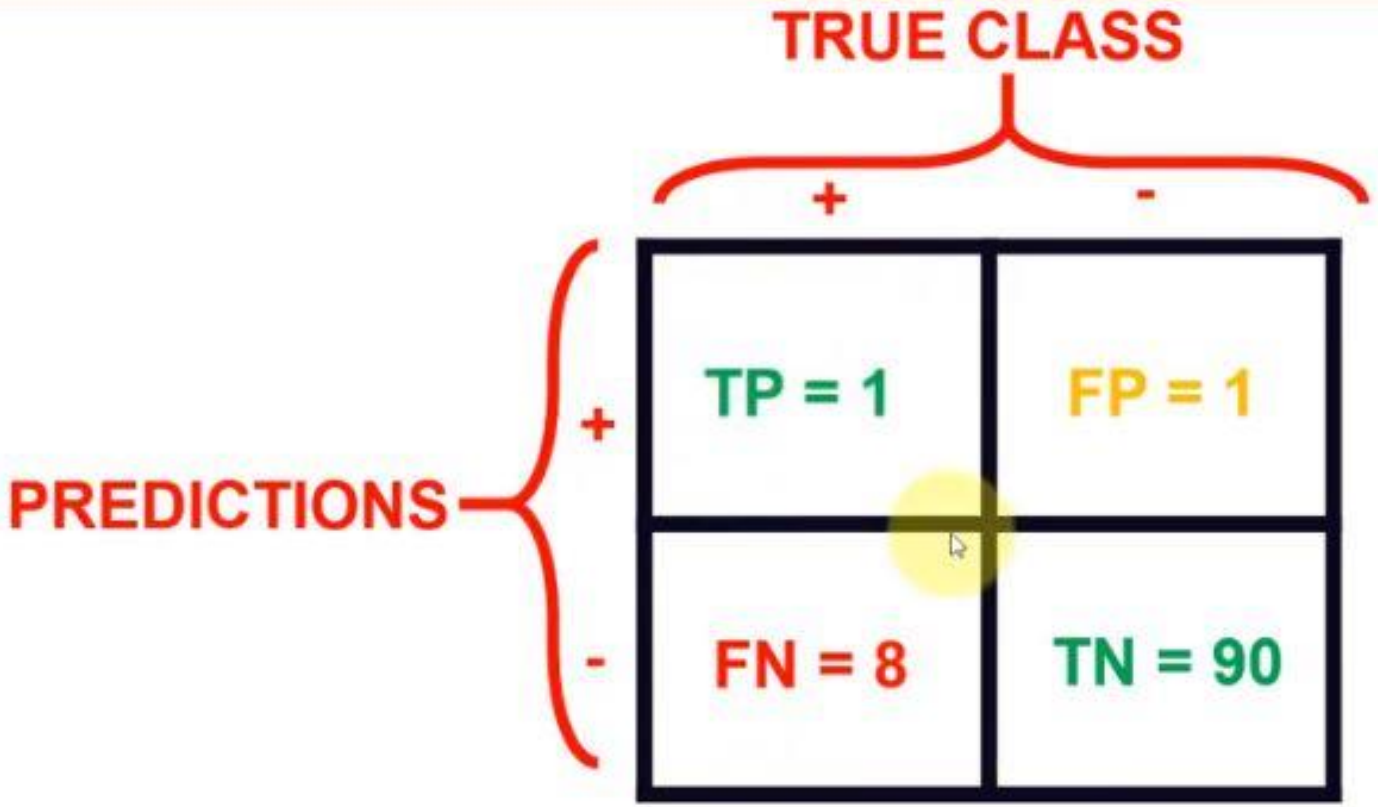
CONFUSION MATRIX

- A confusion matrix is used to describe the performance of a classification model:
  - True positives (TP): cases when classifier predicted TRUE (they have the disease), and correct class was TRUE (patient has disease).
  - True negatives (TN): cases when model predicted FALSE (no disease), and correct class was FALSE (patient do not have disease).
  - False positives (FP) (Type I error): classifier predicted TRUE, but correct class was FALSE (patient did not have disease).
  - False negatives (FN) (Type II error): classifier predicted FALSE (patient do not have disease), but they actually do have the disease

KEY PERFORMANCE INDICATORS (KPI)

- Classification Accuracy = (TP+TN) / (TP + TN + FP + FN)
- Misclassification rate (Error Rate) = (FP + FN) / (TP + TN + FP + FN)
- Precision = TP/Total TRUE Predictions = TP/ (TP+FP) (When model predicted TRUE class, how often was it right?)
- Recall = TP/ Actual TRUE = TP/ (TP+FN) (when the class was actually TRUE, how often did the classifier get it right?)

PRECISION Vs. RECALL EXAMPLE



- Classification Accuracy = (TP+TN) / (TP + TN + FP + FN) = 91%
- Precision = TP/Total TRUE Predictions = TP/ (TP+FP) = ½=50%
- Recall = TP/ Actual TRUE = TP/ (TP+FN) = 1/9 = 11%